Regional Characteristics of Rainstorm Durations and Intensities

by

Conan Lee Hom

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in Civil and Environmental Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1998

Author ....
Department of Civil and Environmental Engineering
January 17, 1998

Certified by ....
Dara Entekhabi
Professor
Thesis Supervisor

Accepted by ....
Joseph M. Sussman
Chairman, Departmental Committee on Graduate Students

Regional Characteristics of Rainstorm Durations and Intensities

by

Conan Lee Hom

Submitted to the Department of Civil and Environmental Engineering
on January 17, 1998, in partial fulfillment of the
requirements for the degree of
MASTER OF SCIENCE
in Civil and Environmental Engineering

Abstract

Rainstorm intensity (i_r) and duration (t_r) strongly influence precipitation partitioning into evaporation, infiltration, and runoff. The i_r and t_r of a storm heavily influences the generation of flood runoff. The time-to-ponding (t_p) in rainfall-runoff transformation is directly dependent on such storm characteristics. Derived distribution of flood probabilities are affected by the joint distribution of i_r and t_r. It is generally assumed that i_r and t_r are statistically independent. A few studies assume a constant negative correlation. In this research, it is shown that there are feasible and infeasible regions or envelopes in the i_r-t_r distribution space. Furthermore, the enveloping surfaces separating these regions may be related to simple observable hydrometeorological variables. This projects examines the influence of readily observed variables upon the envelope of rainstorm duration and intensity combinations.

Thesis Supervisor: Dara Entekhabi
Title: Professor

Foreword

"nil igitur fieri de nilo posse"—Lucretius (de Rerum Natura, 1.205)

I thank my advisor Dara Entekhabi and my colleagues Bruce L. Jacobs (S.M. '87), Susan J. Brown, Amy C. Gieffers ('97), Steven A. Margulis (S.M. '98), Judah Cohen, and the Athena On-Line Consulting Group for their expertise and assistance.

This thesis represents the product of my six years at M.I.T. as an undergraduate and graduate student. I would like to recognize those who have been part of the experience: the gentlemen and brothers at the EΘ Chapter of ΣN Fraternity Inc. (most notably the mathematicians), David M. Frohman ('95), Matthew P. Mello, Monica K. Black, Amy L. Arnold, Rebecca J. Hill ('95), Michael V. DiBiasio, E. Gordon Hamilton, Gregory D. Barringer, Kimberly A. Walsh, Jeffrey P. Leonard, Veronica B. Culbert, Paul D. Fricker, Dr. T. Francis Ogilvie (Professor Emeritus), Deirdre K. Dunn ('99), and to the many jazz musicians to whom I listened during those long sessions on the computer.

A special recognition goes to Dr. Jon "Ted" Lendon of the Corcoran History Department at the University of Virginia and of the history department at the Massachusetts Institute of Technology, and to "Young" Kyle D. Welch ('90). Not only an accomplished sailor, but also a backgammon ace, Kyle demonstrated some probability concepts to me on a backgammon board in the Commodore's room. These two gentlemen heavily influenced the way I approached writing this thesis. One evening at the Muddy Charles Pub, they asked me three simple questions—*Why? So what? Why should we care?* I have tried to answer those questions amidst the theory and accompanying hullaboo.

The development of this thesis has been supported by a fellowship from the National Aeronautics and Space Administration (NASA) research grant NAS 5–31721 and from a National Science Foundation (NSF) graduate fellowship.

Conan L. Hom
Saturday, January 17, 1997
Lexington, Massachusetts USA

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

From a practical engineering perspective, most accepted hydrologic techniques have focused on the fate and transportation of water from the moment it touches the ground to its final destination, whether it be to a sink such as the ocean or a lake, or directly back into the atmosphere where it becomes precipitation again at a later time. To humans such focus makes sense. When water is on the ground, it directly affects the livelihood of the inhabitants. There, man can harness it usefully by various means and also control or at least minimize its destructive traits, unlike when it is in its vapor form in the atmosphere. The performances of the various hydrologic techniques, however, are critically dependent upon the input characteristics of water into the system which is being modeled. Over land, the primary input source of water is precipitation. Precipitation is grouped temporally into a series of rainstorms.

A rainstorm delivers water to the ground and, depending on the ground's absorbtion characteristics, the near surface atmospheric state, and the local terrain features, the arriving water suffers or, using the more hydrolgic expression, is *partitioned* into three general fates of varying quantities: Water evaporates into the air, infiltrates the soil, or forms surface runoff. The surface runoff is of utmost immediate importance to humans for the purposes of drought prevention, irrigation, and flood control. The amount of runoff generated during a given time period can have major environmental and societal effects which last for years. Evaporation, however, cannot be considered trivial. Its measurement is a major contributer to accurate weather forecasting. Water transfers significant amounts of energy into the atmosphere because of its high latent heat of evaporation. At the surface, evaporation rates are heavily influenced by the degree of saturation of the soil. Water that infiltrates the surface is available for aquifer recharge or influences subsurface groundwater flows.

If one considers the local ground properties as fixed, and near surface atmospheric conditions as steady or at least easily measurable over a wide basis, then rainstorm intensities and durations—the itermittency characteristics of preciptiation events—become the primary variables in the partitioning of the incident precipitation [19]. An intense storm may overcome the infiltration capacity quickly and the balance of the water input may become runoff. Likewise, the input from a long, low duration storm may infiltrate mainly into the ground. The time between storms also can influence the infiltration rates. The longer the time between storms, the longer the the ground surface has to reduce its saturation state and 'recover' its infiltration capacity. The recovery takes place through reevaporation, and/or by percolation to deeper ground storage.

The reader is referred to a more in depth approach to infiltration-runoff generation by Bras [2]. While an extensive analysis is avoided here, the basic elements of the approach follow [10].

1.2 Infiltration and Runoff

One of the long standing equations for determining infiltration of water from the surface is the Phillips equation. This equation performs well for short time solutions. Without statements of the mathematical proofs, the basic equations and results derived from the Philips equations are displayed below. What is important, and therefore discussed, is their implications.

The Phillips equation is:

$$i(t) = \sum_{t=-1}^{\infty} A_n t^{n/2} \tag{1.1}$$

where $i(t)$ is the infiltration rate per unit area [L/T] from a ponded surface, and each A_n is a constant. In general analysis and applications, the first two terms are the most important due to rapid convergence [28, page 243]. Therefore, the Phillips equation simplifies to

$$i(t) = S_i t^{-1/2} + A_\circ \tag{1.2}$$

where S_i is the constant A_{-1} and stands for the soil sorptivity. The constants can be determined by initial conditions of the soil and the soil hydraulic model [3]. While the Phillips equation measures infiltration from a ponded surface it is possible to use the equation to model infiltration from a soil surface that is not initially ponded. Such conditions occur often at the beginning of a storm.[1] The critical parameter to determine is the time-to-ponding (t_p). After creating a water balance between the Phillips equation and the incoming water (denoted by the rainfall intensity i_r), the Phillips curve is shifted by a time compression factor t_c [27]. The time to ponding is determined by the intersection of the shifted Phillips curve and the intensity of the storm. The time to ponding

$$t_p = \frac{S_i^2}{2i_r(i_r - A_\circ)}\left(1 + \frac{A_\circ}{2(i_r - A_\circ)}\right) \approx \frac{S_i^2}{2(i_r - A_\circ)^2} \tag{1.3}$$

reflects the time it takes from the onset of the precipitation to reach a state where the ground can no longer absorb all the incoming water. t_c is also related to t_p

$$t_c = t_p + \left(\frac{S_i}{2(i_r - A_\circ)}\right)^2 . \tag{1.4}$$

At time t_p, the surface reaches ponded conditions and infiltration behaves like the pure Phillips curve from time t_p onwards as long as precipitation input is adequate to maintain the ponded conditions at the surface. After time t_p, some of the incoming water still infiltrates the surface but the rest of the water gathers on the surface and

[1] Note that evaporation is not considered in this analysis. During a rainstorm evaporation is effectively zero (with the exception of the rainless periods of a storm).

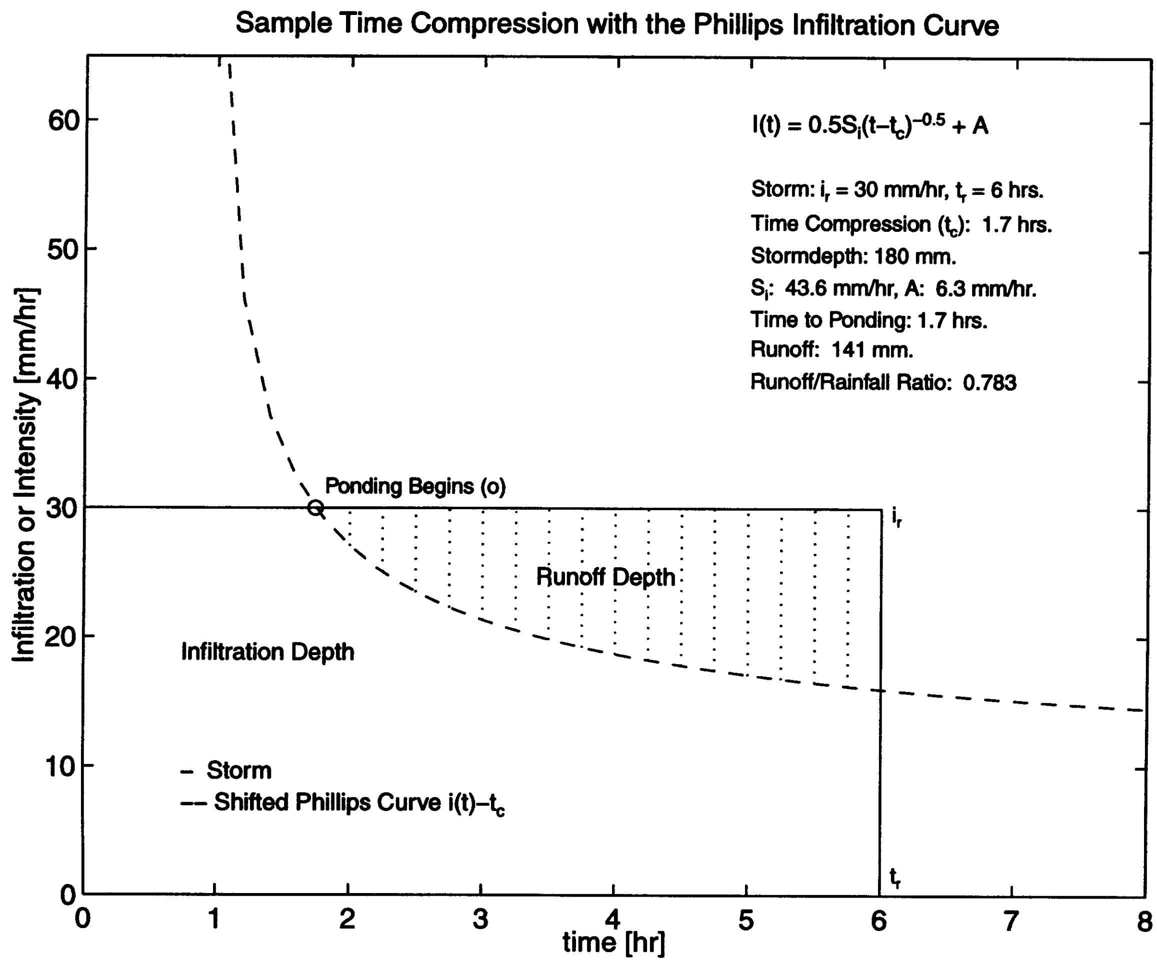

Figure 1-1: Philips curve infiltration and run-off generation.

forms run-off which is calculated as

$$R_s \approx (i_r - A_\circ)t_r - S_i(t_r/2)^{1/2} \tag{1.5}$$

where t_r is the rainfall duration.

Note that in Equations 1.2 and 1.3, the run-off R_s and the time-to-ponding t_p depend on a storms intensity i_r and duration t_r, as well as on the soil characteristics. Figure 1-1 shows a graphical example of the Phillips infiltration and run-off generation. In order to estimate the probability distribution of run-off, it is necessary to use the the joint distribution of rainstorm intensities and duration [10].

1.3 Current Technologies for Observing Precipitation (and Limitations)

As a general rule, to achieve effective and accurate hydrologic techniques that are applicable to everyday life, precipitation data are needed. Measurement of precipitation is a more difficult task than it seems to the layman. Today, however, recent technological advances have permitted more widespread precipitation measurement methods.

Characteristics of precipitation fields change depending on the spatial scale. At the microscale, from a few hundred meters to several kilometers, rain cells and convective updrafts are dominant. At the mesoscale, from tens to hundreds of kilometers, rainbands and convective complexes are present. Even larger is the synoptic scale at which one can describe general weather systems. Within these various spatial scales, regions of temporal pulses of higher and lesser rain intensity are also embedded.

1.3.1 Raingauges

Raingauges offer point and time-accumulation measurements of precipitation that reaches the ground. Unfortunately, due to wind patterns around a raingauge, there is some accuracy loss due to the deflection of the raindrops. When the precipitation is snowfall, accuracy drops significantly with wind speed [2, 11]. Due to cost and practical reasons, a raingauge network of sufficient size and density to characterize rainstorms across all of its spatial and temporal scales impossible to maintain or operate. Raingauges also have some locational bias. They tend to be located nearer to settled areas.

The above is not to imply that raingauges are without benefits. Of the current technologies in use, raingauges are the oldest and thus have a wealth of historical precipitation data. The data can be used for historical analysis of precipitation as will be seen in following sections. They provide a callibration basis for and supplement the following methods.

1.3.2 Radar

Recent advances have made radar an important tool for precipitation monitoring and measurement [12]. Raytheon's NEXRAD is one of the newest technologies available. The micro-wave doppler radar sweeps about 60 degrees a minute. Although the micro-wave nature of its signal requires a significant power supply [14], it can provide fine scale reflectivity data up to several hundreds of miles. Reflectivity, or radar echoes, indicate the presence of hydrometeors and hence rain. Almost the entire United States is covered by NEXRAD stations. NEXRAD does suffer the traditional ground radar problems. First of all it is limited by its line-of-site (LOS). Mountains and curvature of the earth are the traditional LOS limits [14]. Also, ice crystals have greater reflectivity than raindrops, with the result that precipitation is overstated. Airmasses deflect signals. The ability to measure precipitation behind another field of heavy precipitation is hampered by signal attenuation. To avoid ground clutter, the radars are tilted upwards thus ground coverage extends out to seventy miles. As a result there is a range bias with distance. At longdistances, NEXRAD may indicate precipitation that reevaporates long before reaching the ground.

As with raingauges, world coverage under NEXRAD is currently impractical (again like over oceans) and expensive. It must be stated that under favorable circumstances NEXRAD provides the best, least biased precipitation information available. In these conditions it can provide hourly or even less than hourly precipitation data.

1.3.3 Satellites

The major advantages of satellites is their ability to observe areas of the earth that would be impractical to measure by raingauges or radar, and to observe the earth on a useful areal resolution. Satellites measure both microwave emissions, and infrared/visible emissions and reflections from the Earth. Unlike rain gauges and radar, they can observe large continuous portions of the Earth at a time. Microwave channels provide accurate measurements of precipitation over oceans and surfaces with known emissivity. Unfortunately current technology is unable to make the microwave sensing devices small enough to be put into a geosynchronous orbit thus, with the result that, in polar orbits, microwave systems can only measure any given area a few times a day at most [16]. Infrared systems can be placed into geosynchronous orbit and provide continuous data. Infrared, however, gives less accurate estimates because the retrieval algorithm relies on the weak correlation between cloud depth and rainfall rate. The Global Precipitation Climatology Project (GPCP) [15, 16] combines multispectral emisions and reflections measurements to provide accurate areal monthly averages or mean hourly precipitation over a month.

1.4 Precipitation Models

Satellites provide a wealth of precipitation data but not on the finer temporal scales needed to facilitate accurate hydrologic analysis. Although it may be impossible to actually disaggregate the satellite monthly means into the real hourly precipitation time-lines, other options may be available. A next best solution may be to create a statistically viable synthetic precipitation time series where the statistical characteristics of the actual hourly rainfall are preserved. In current analysis of precipitation, various studies are concerned with the mean, variance, first-lag autocorrelation, and the probability of rain [11, 26]. By matching the statistical characteristics, it is hoped that the general distribution of rainfall throughout the month can be imitated. For the purposes of analysis, testing of methods to generate synthetic series can be done against actual point rainfall measurements from rainguages.

An important requirement of such a method is that it can be applied to unmeasured areas. The ideal synthetic precipitation series creation method has two goals:

1. The synthetic series approximates the statistical properties of the area even though the areal statistics are never actually measured,

2. The synthetic series retains a sufficient amount of reality. That is, the synthetic precipitation events are realistic and feasible for the area in question.

There is, therefore, a motivation to have at least part of the method based on a regional parameter(s) which reveals and contains the essential information rather than a long actual precipitation history which, from the above discussion, would be impractical to find. Such a parameter(s) would be good if

- It is easily observable (such as from a satellite) on a world-wide basis, The parameter can be observed on a small enough scale for precipitation modeling, and,

- The real time parameter value circumvents the need for observed precipitation time series.

The approach of this paper assumes that there are easily observed parameters which, at least, can help define the set of all possible storms in an area, and that these parameters, for the most part, do satisfy the two requirements listed directly above.

Assuming that the arrival of independent rainstorms is a random process, and given a total monthly rainfall (such as that observed by satellites), one possible method for generating a synthetic precipitation series is to randomly sample off a rainstorm intensity-duration combination distribution until the total depth of all the rainstorms that are selected from the distribution equals the total monthly rainfall. Then each of the selections of storms is placed randomly along the time period, so that a synthetic times series is created.

One way to order the probability of various storms is to create a cumulative distribution according to total depth of a storm. Through intuitive reasoning, high depth (total rainfall) storms, which deplete the precipitable water in the local atmosphere, may be less likely to occur. An issue that arises with the cumulative distribution method is how to assign storm and duration intensities to the individual depths since it is quite obvious that different intensity-duration combinations can yield the same depth. That issue along with how these storms, once selected, are placed to preserve storm independence are beyond the scope of this thesis.

A classical engineering concept is related to the results presented in this study. *Intensity-Frequency-Duration* (IFD) curves [2, 32] are commonly used in hydraulic structure designs. Based on historical records and regionalizations, IFD curves allow the selection of consistent storm intensities and durations that have a given return period or probability of recurrence. The development of the iso-return curves and its related sampling methodology are beyond the scope of this thesis although it is noted that Cao [4] suggests the existance of regionally based homogeneities of IFD curves. Roughly, one would have to develop an algorithm for choosing a storm on the same iso-return curve, once the return period is specified during the sampling process. Nevertheless, the IFD concept is more desirable than the depth determined cumulative distribution because the IFD curves do not rely on the intuitive but unconfirmed assumption that higher depth storms are always more unlikely than lower depth storms. In contrast to IFD curves where combinations of storm intensities and durations are selected and sorted according to their return period, the PPF envelopes developed in this thesis mark the boundaries between the feasible and infeasible storm intensities and duration combinations. Within the envelope (or feasible region) and in the joint probability distribution space, such storm characteristics may be sampled for event-based hydrologic analysis. In addition the envelopes define the regions in which the IFD curves must lie.

Before one can start sampling off any rainstorm distribution, the set of possible storm duration and intensity combinations must be found, and their individual probablities assigned. This thesis engages the former topic. It also explores whether or not there exists some easily observed, real time regional parameters that can reveal the set of all possible rainstorm duration and intensity combinations for a given area,[2] and, if such parameters exist, what they are. The motivation for finding such parameters is that the primary historical sources are from raingauges. As stated before, they are point measurements. Radar observations too, do not have complete coverage and radar observation history is even shorter than that of raingauges.

[2]Similar to Cao's [4] suggestion of regionally based IFD curve homogeneities.

Chapter 2

Project Description

2.1 Preview

A brief preview is provided to help the reader with the purpose and structure of the thesis. As stated in Section 1.4, the thesis seeks to define the set of likely storm intensities and storm durations for a region. Examining Figure 2-1, it is assumed that an envelope exists which bounds the set of possible storm intensity and duration combinations from the set of infeasible, extremely improbable and or impossible ones. This line is denoted as the rainstorm *production possibility frontier* (PPF). The concept will be elaborated later in this Chapter. For now it is simply stated that the PPF may be characterized by factor inputs and a general functional form. To test various factors and functional fits, a database of storms must be provided. Therefore one of the key primary steps is the identification of independent rainstorms.

2.2 The Rainstorm Event

2.2.1 Definition

A rainstorm is a precipitation time series in which the elements of the precipitation series are non-independent and/or related in some manner. Within the rainstorm, the precipitation may vary in intensity. In reality, it may be raining for a continuous

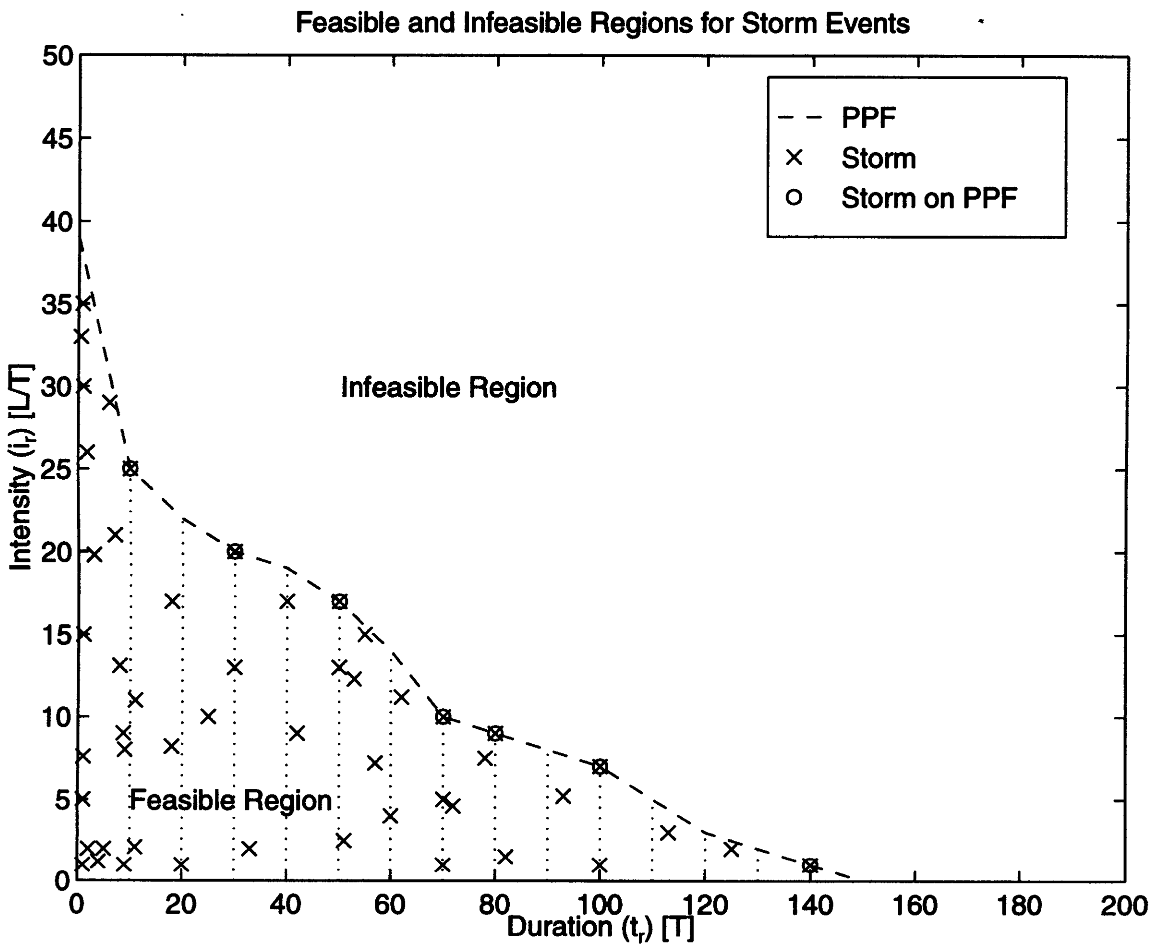

Figure 2-1: Feasible and infeasible storm event i_r-t_r combinations.

period of time, but, as common experience shows, at certain times, it pours and at certain times it drizzles. Sometimes within an event, the precipitation series is intermittent. There occur rainstorms in which for periods of time there is no precipitation but the precipation that bounds both ends of the no rain time period are still part of the same physical storm system.

A certain minimum time period of no rain must pass before the next precipitation event arrival is generally considered to be statistically independent of past events and part of another rainstorm. This minimum time length is T_{bmin} [T]. During a continuous rainless time of length equal to or greater than T_{bmin}, one storm system passes, and then the next system moves in. When the two precipitation series are separated temporally by at least T_{bmin} of no rain, the precipitation series of one rainstorm is considered to be independent from that of the another rainstorm.

2.2.2 Terminology

With the basics of an independent rainstorm now defined, the terminology is now fully introduced. The total time length or duration of a rainstorm is defined as the time from its first precipitation pulse to its last, is known as t_r [T]. The total volume of precipitation of the rainstorm event per unit area is the storm depth h [L]. The time between the time origin of two sequential independent storms is t_b [T]. For storms characterized as rectangular pulses (rain intensity constant for any storm), the average rainstorm intensity of a storm denoted as i_r [L/T] is the storm depth divided by the duration ($\frac{h}{t_r}$). Because the temporal intensity changes within a rainstorm are difficult to characterize, an individual rainstorm can be and is often described as a rectangular precipitation pulse of length t_r, (average) height i_r and an total area or depth of $h = i_r t_r$. This concept is known as the *Rectangular Pulse Model* [25, 26]. Figure 2-2 shows the concepts of i_r, t_r, and t_b.

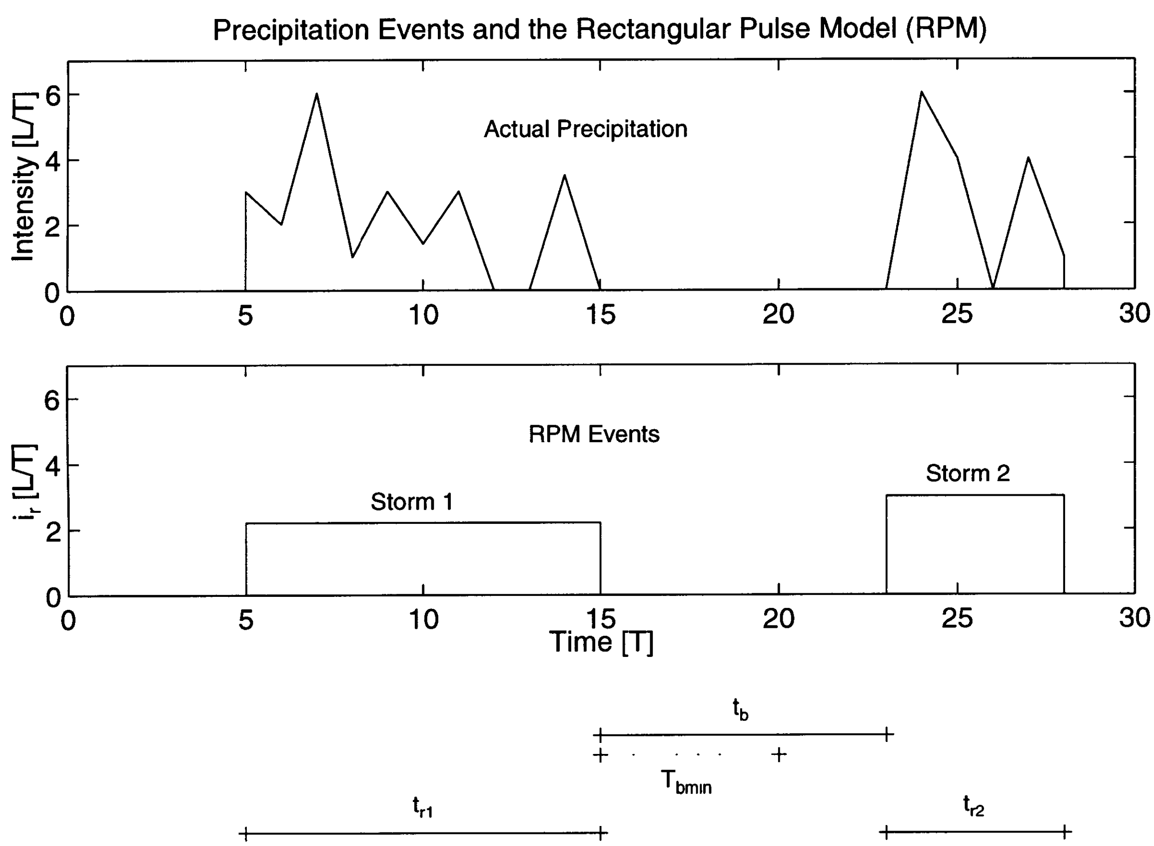

Figure 2-2: i_r, t_r, t_b, and T_{bmin} representing storm rectangular pulses over time.

2.2.3 Poisson Arrival Process

Eagleson [7, 8, 9] describes that the number of independent rainstorms within a time period is characterized as a Poisson distribution, where the probability of x rainstorms within a time period t is

$$f(x|\omega t) = \begin{cases} \frac{(\omega t)^x e^{-\omega t}}{x!} & \text{for } x = 0, 1, 2, 3, 4, ... \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

and ω is the average arrival rate of independent storm events. An important property of the Poisson distribution is that both the expected value and variance are equal. In this case, the expected value and the variance are equal to ωt where ωt can be interpreted as the average number of storms for the time period of length t.

If indeed rainstorms are a Poisson process, then the Poisson distribution implies two assumptions about rainstorms [6, pages 252–257].

1. The numbers of storms in any two separate and nonoverlapping time intervals are independent of each other. That is, the number of arrivals in time period A do not give any information about the number of arrivals in time period B [31, page 1108].

2. The shorter a time interval is, the lower the probability is of a rainstorm occurance in that time interval. In fact, the probability is approximately proportional to the time interval length.

2.2.4 Storm Identification

Most relevant to this thesis is the fact that, if rainstorms do behave in a Poisson arrival process with a parameter ωt, the elapsed time between the beginning of one storm to the beginning of another storm, also known as the *interarrival time*, is exponentially distributed [6, 9, 31] with parameter ω,

$$f(t_a|\omega) = \begin{cases} \omega e^{-\omega t_a} & \text{for } t_a > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

where t_a is the interarrival time. This distribution may be deduced from Equation 2.1 given no arrivals ($x = 0$) over the period between storms (t_a). One property of the exponential distribution that will be used later on is that the expected value ($E[t_a]$) and the standard deviation ($\sqrt{Var[t_a]}$) are equal [6, 24, 33]. From this property one concludes that the coefficient of variation (CV) of an exponential distribution, which is defined as

$$CV = \frac{\sqrt{Var[t_a]}}{E[t_a]}, \tag{2.3}$$

is unity.

With an additional assumption it is possible to use this concept of the interarrival time to indentify individual independent rainstorms from a precipitation timeline [24, 33]. If we assume that

$$\omega m_{tr} \ll 1 \tag{2.4}$$

where m_{tr} is the average storm duration, then the time between the end of a rainstorm to the beginning of the next rainstorm (t_b) is also exponentially distributed

$$f(t_b|\beta) = \begin{cases} \beta e^{\beta t_b} & \text{for } t_b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

where β is an undetermined parameter representing the inverse of both $E[t_a]$ and $\sqrt{Var[t_a]}$. β, in general, does not have to equal ω because the distribution of the storm durations is uncertain and may differ from the arrival process.

Knowing how t_b is distributed, enables us to identify the independent storms. The general idea is to find a minimum timelength T_{bmin} for which if a rainless time period is equal or greater than T_{bmin}, the next period of precipitation is considered a new, (therefore) independent, storm. The important assumption for the two methods described below is that t_b is, in actuality, exponentially distributed.

Coefficient of Variation

One such method of finding T_{bmin} is through the coefficient of variation method [24, 33]. Since it is assumed that t_b is exponentially distributed, it follows from the property of the exponential distribution (Equation 2.5), that $E[t_r = \sqrt{Var[t_b]}$. The coefficient of variation (CV) of t_b is equal to one.

$$CV[t_b] = \frac{\sigma}{\mu} = \frac{\beta}{\beta} = 1. \tag{2.6}$$

To determine T_{bmin}, given a precipitation timeline history of region, one begins by eliminating rainless time intervals as short as time $\triangle t$—the shortest rainless period in the precipitation series. The sample CV of the remaining rainless intervals is calculated and checked if arbitrarily close to one. If not, the process is repeated again. This time all the rainless periods less than $2\triangle t$ are eliminated and the sample CV is again calculated and checked if equal to one. The process is repeated over and over again and the current $\triangle t$ increased one step each iteration until the sample CV is close enough to or equals one. When that point is reached, $T_{bmin} = n\triangle t$, where n is the count of time periods. One of the draw backs of the CV method is that it is quite possible that the rainless periods within a storm are also exponentially distributed. Therefore, during the iteration process, CV may reach the value of one prematurely. As a result, the CV method may give a T_{bmin} that is too small. It is for this reason that the Breakpoint method [33], an alternative to the CV method for independent storm identification, is used in this study.

Breakpoint

Since the rainless periods equal or greater than the actual T_{bmin} are exponentially distributed, the semilog plot of the cumulative probability of those rainless periods should be linear. The rainless periods less than T_{bmin} can have another distribution, or even another exponential distribution (with storms) but as long as exponential distribution has a different parameter, the slope of the cumulative probability will change once intervals of T_{bmin} or greater are evaluated. The breakpoint method divides the cumulative probability plot of rainless periods into two pieces based on a two-phase regression in semilog space. The optimal division is determined when the total error sum of squares for the two-phase regression is a minimum. This meeting point of the two pieces—the *breakpoint*—is the minimum time between independent storms. The breakpoint is where the slopes of the cumulative probability distribution changes. The essential idea is that the semi-log linearity of the exponentially distributed time between independent rainstorms is captured in one of the regressions, and the other regression phase captures the behaviour of the within storm rainless periods.

The breakpoint method preserves the possibility that the rainless (intermittent) periods within a storm are arbitrarily distributed. However it cannot rule out if the rainless periods within and between storms are identically distributed, but it is an opinion of the author that this is unlikely.

Using the precipitation data, and the breakpoint method, individual storms were identified at one hundred sixty one rain-gauge stations across the United States of America.[1] The characteristics of an individual storm that were recorded were the individual storm depth h [mm], its average intensity i_r [mm/hr], and duration t_r [hr]. One observes that the duration captures the total time of the storm *including* the rainless periods within, and that the average intensity, the i_r of the storm, is simply the total storm depth divided by the duration. In accordance with the Rectangular Pulse Model, each storm has one intensity level.

Figures 2-3 to 2-6 show the storm event breakdown using the breakpoint method for the summer season (June-August) at Apalachicola FL, Boston MA, Huntsville AL, and Nashville TN. Note that generally intense storms (large i_r) are necessarily of short duration (small t_r) but low-intensity storms such as drizzles (low i_r) may be of either long or short durations. Therefore, the relationship between i_r and t_r is neither independent, nor does it have a negative correlation. Instead, an envelope separates the physically possible i_r-t_r combinations from the the generally infeasible combinations. Now that the independent rainstorm characteristics have been extracted from

[1]See Subsection 2.3.4 for discussion on the data.

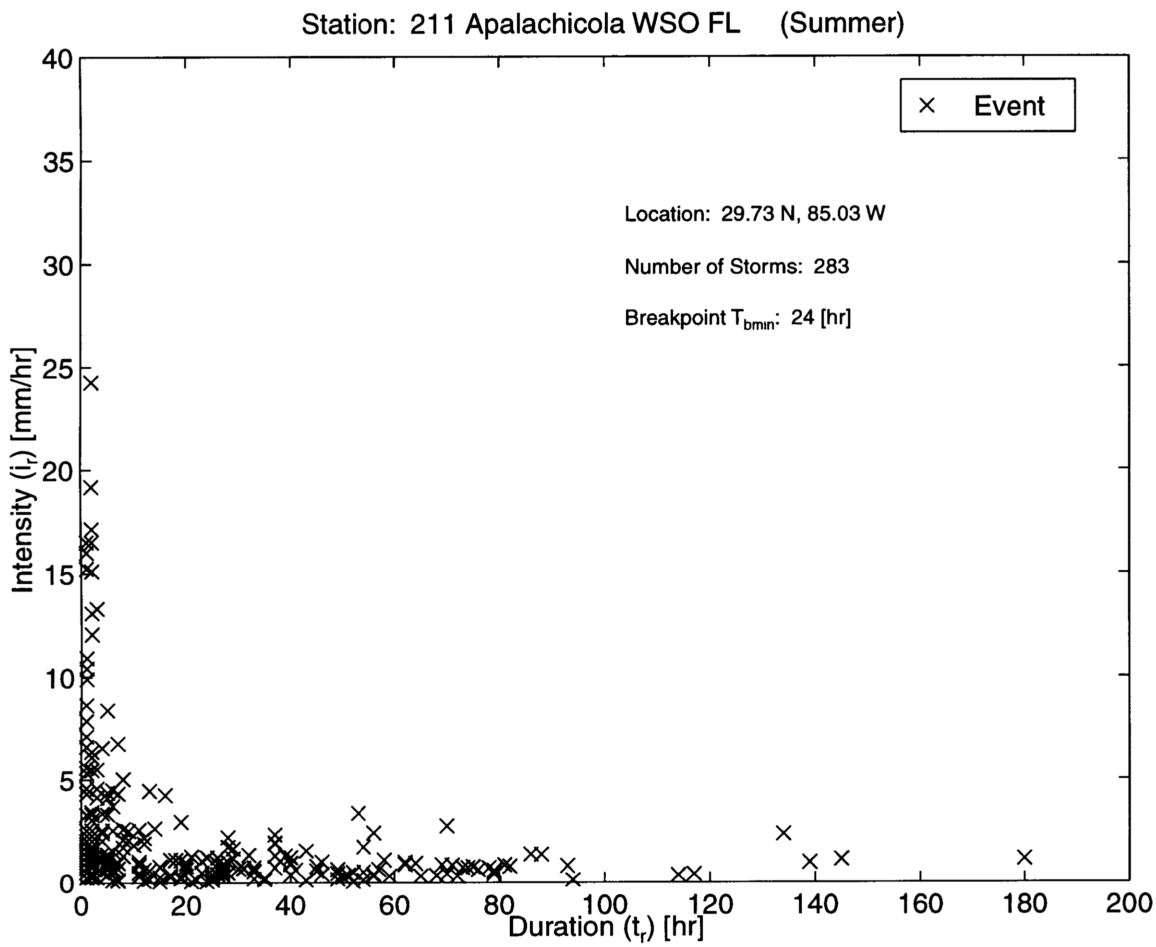

Figure 2-3: Breakpoint determined storm event i_r-t_r combinations over a fifteen year period (1971–1985) at Apalachicola, FL.

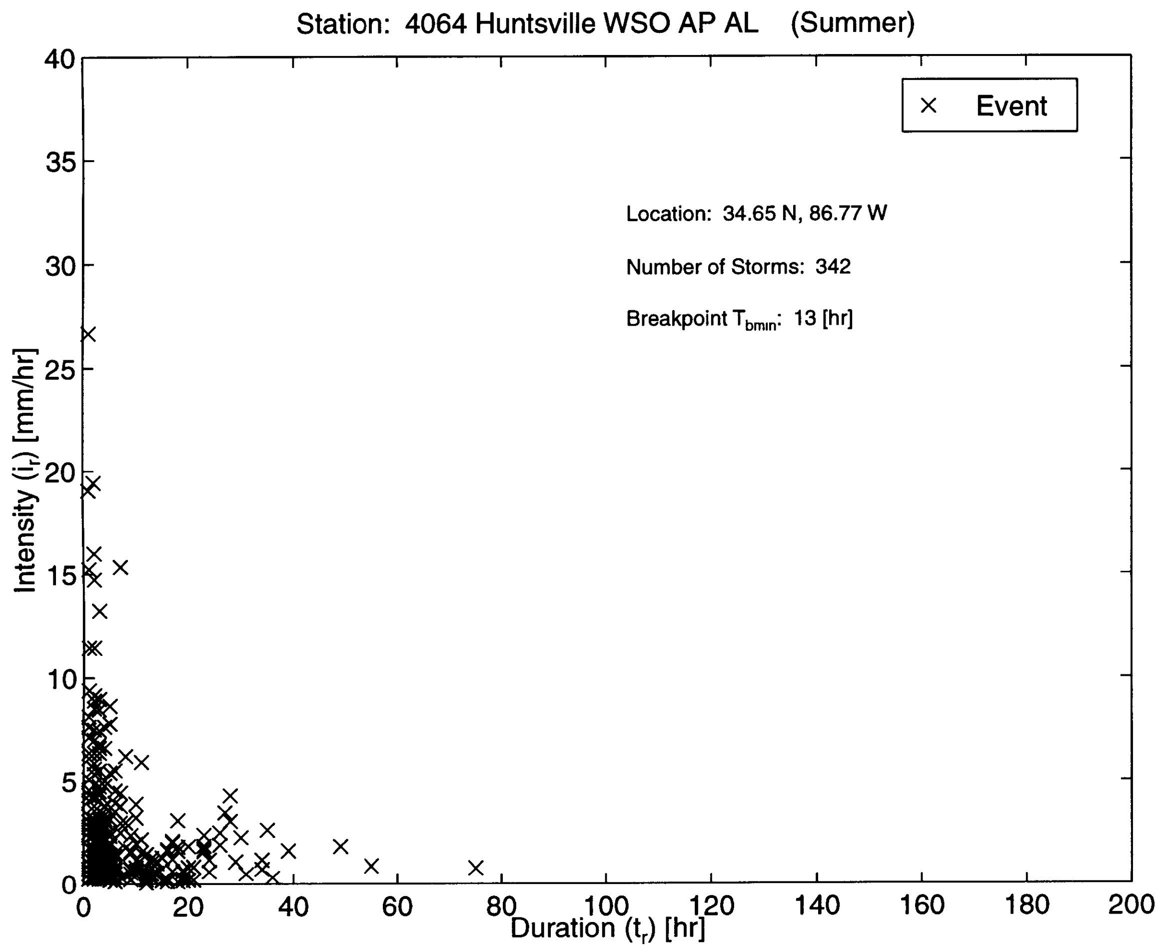

Figure 2-4: Breakpoint determined storm event i_r-t_r combinations over a fifteen year period (1971–1985) at Huntsville, AL.

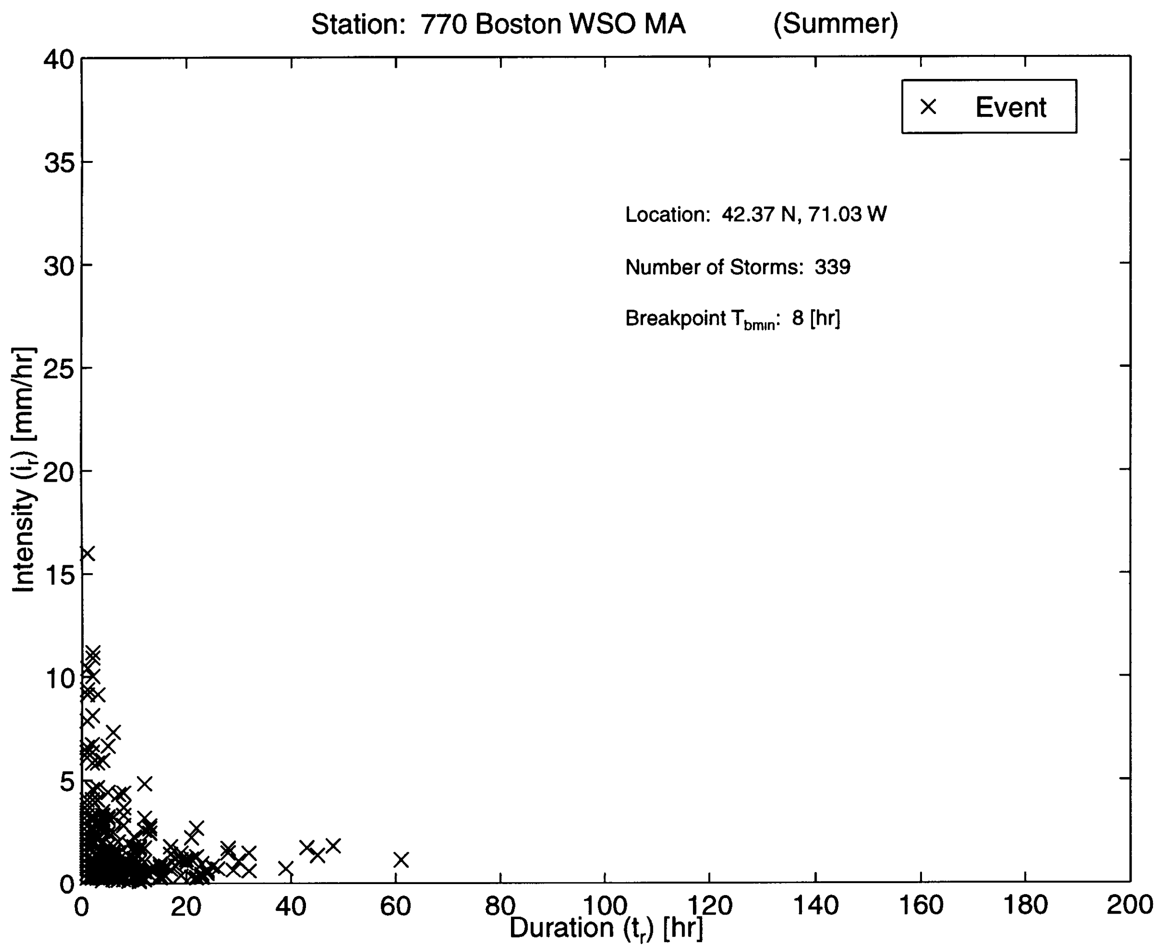

Figure 2-5: Breakpoint determined storm event i_r-t_r combinations over a fifteen year period (1971-1985) at Boston, MA.

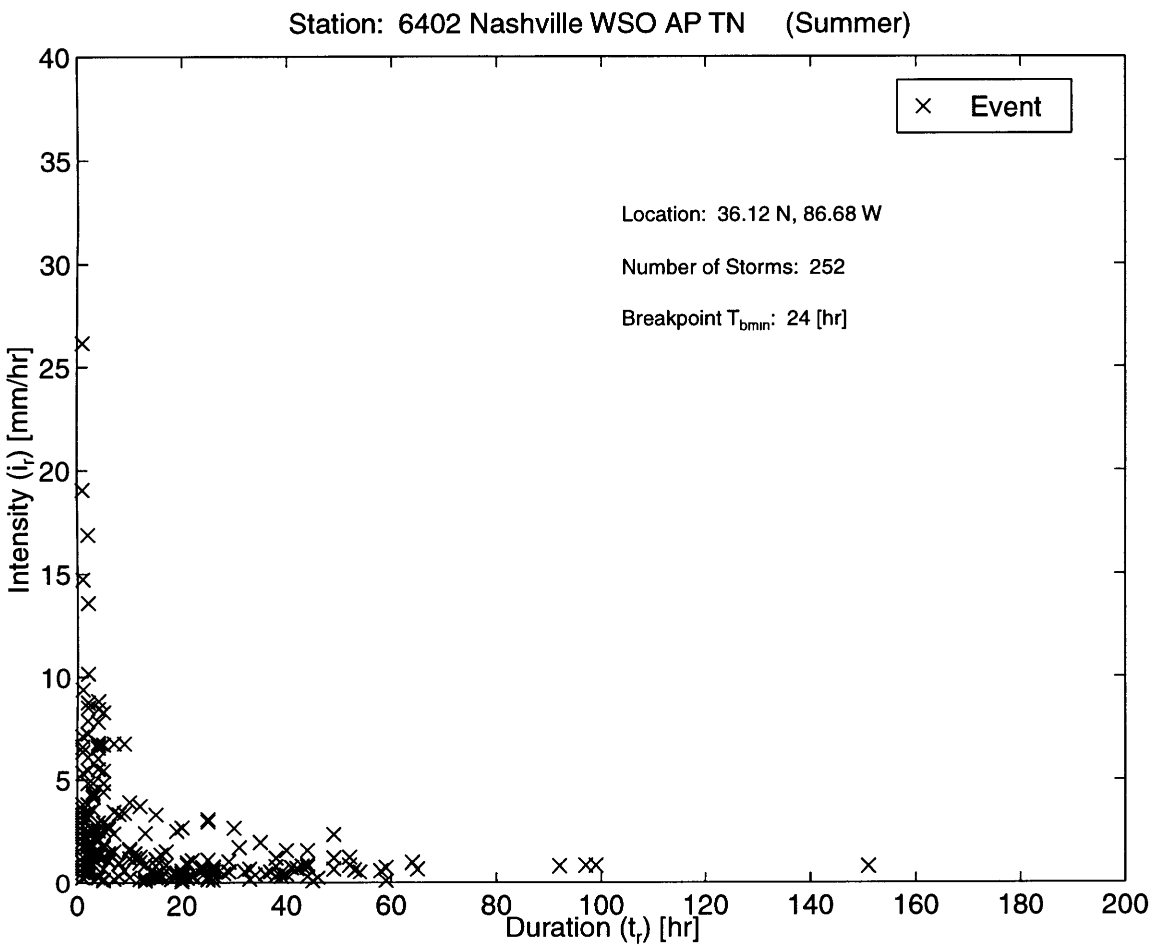

Figure 2-6: Breakpoint determined storm event i_r-t_r combinations over a fifteen year period (1971–1985) at Nashville, TN.

precipitation timelines in a consistent manner, we can proceed to define the envelope for the feasible set of storms.

2.3 Possible Rainstorm Combinations

2.3.1 Previous Models

Earliest assumptions about possible duration-intensity combinations [9, 26] considered that i_r and t_r were both exponentially and independently distributed

$$f(i_r) = \mu e^{-\mu i_r} \tag{2.7}$$

$$f(t_r) = \eta e^{-\eta t_r} \tag{2.8}$$

where μ and η represent the inverses of the expected values of i_r and t_r respectively.

Several unrealistic implications result from this assumption. First, from the assumption that t_r and i_r are independent, all i_r-t_r combinations are possible. This cannot be universally true from an physical perspective. Intense storms cannot be long. High intensity and duration storms may be more improbable than indicated by a joint probability distribution of Equations 2.7 and 2.8. For independent random variables, their joint distribution is the product of their marginal distributions. Indeed, high intensity and duration storms, if they occur, may have to be driven by, not local and regional parameters, but rather by larger scale "synoptic" influences. According to Bacchi [1], the independence of i_r and t_r assumption is a possible reason as to why the model above performs inadequately in extreme value (of intensities) simulations.

By examining Figures 2-3 through 2-6, one can see that low duration storms can be either low or high in intensity intensity. In addition, long duration storms are as restricted to being low in intensity. The airmass convergence on the scale needed to supply the water vapor for an intense rainstorm cannot be maintained for an extended period of time in a convective storm. In a less intense storm however, the convergence can sometimes be maintained. As a conclusion, there is no guarantee that either i_r or t_r is exponentially distributed, nor is there a likelihood that they are independent of each other.

A second model of possible storm combinations is shown by Bacchi et al. [1, 18]. In this model, i_r and t_r are correlated. The probability distribution function is

$$f_{i_r t_r}(i_r, t_r) = \mu\eta[(1 + \mu\delta i_r)(1 + \eta\delta t_r) - \delta] \exp(-\mu i_r - \eta t_r - \mu\eta i_r t_r) \tag{2.9}$$

where the inverses of μ and η again represent the expected values of i_r and t_r respectively. Analyzing Equation 2.9 closely, one can see the inherent exponential distribution quality of i_r and t_r. δ is a parameter that controls the correlation between i_r and t_r. This parameter ranges from zero to one and is arbitrarily chosen. The correlation coefficient

$$\rho(i_r, t_r) = -1 + \int_0^\infty \frac{1}{1 + \delta y} \exp(-y)\, dy \tag{2.10}$$

is a constant value once δ is specified. ρ ranges from 0 to -0.404 implying that i_r and t_r are essentially negatively and linearly correlated. This model is therefore superior to the independent assumption in that it does attempt to capture some negative correlation. Reconsidering figures 2-3 through 2-6, it is clear that the i_r-t_r relation is not linear, and that, instead, an enveloping curve separates the feasible and infeasible combinations of i_r and t_r.

2.3.2 The Rainstorm Envelope

The storm plots for Boston, Apalachicola, Huntsville, and Nashville (Figures 2-3 through 2-6) one might conclude that the set of possible storms is bound by an envelope. Such an envelope is commonly known as the feasibile region, outside of which the various i_r-t_r combinations are infeasible (see Figure 2-1). Within the feasible region the (linear) correlation between i_r and t_r is not evident as one sees storms all over localized areas. The defining boundary (other than the zero intensity and duration axes) is the storm *Production Possiblity Frontier (PPF)* analogous to its

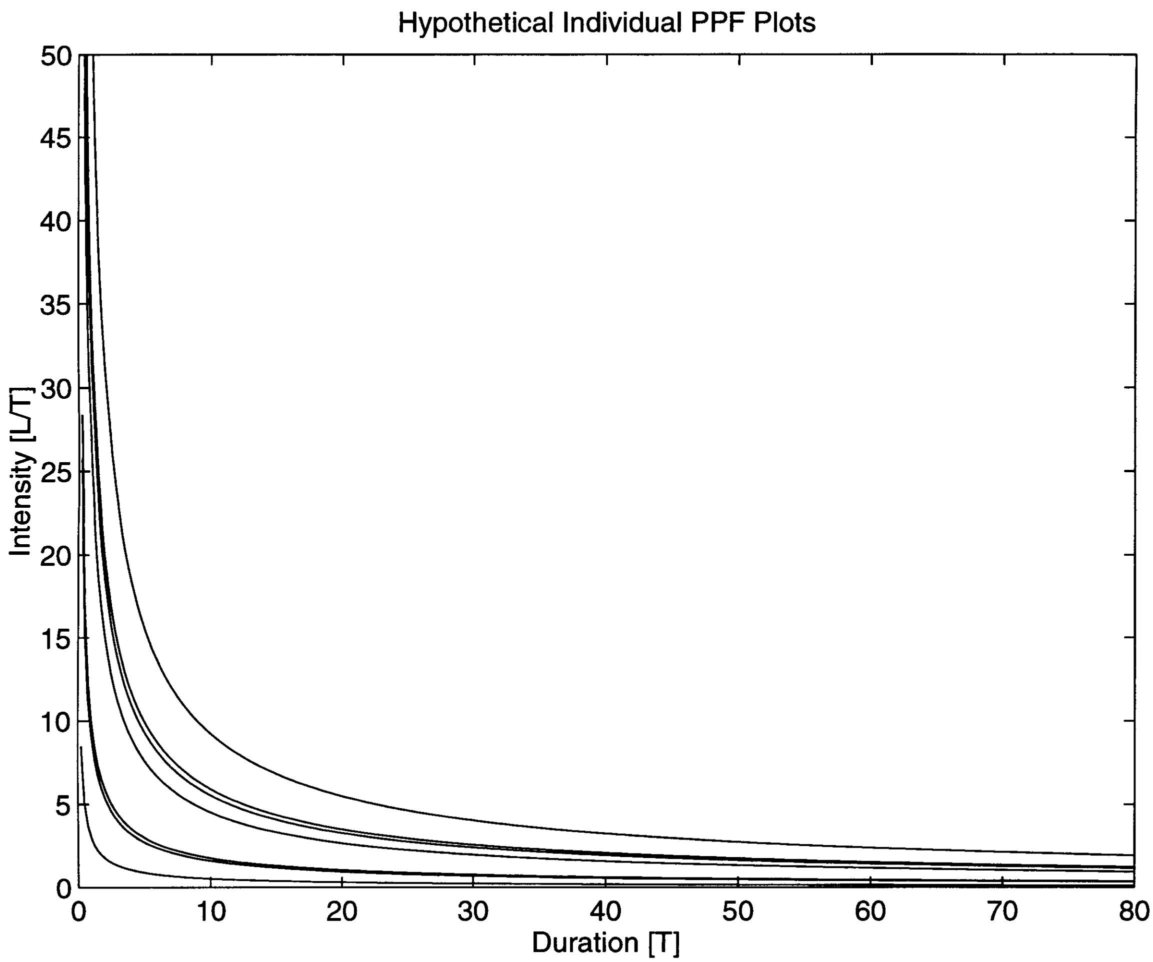

Figure 2-7: Individual Regional PPFs.

economics firm theory counterpart [21]. Below the PPF lie all the possible or feasible storm combinations.

Recalling the model to produce viable synthetic precipitation timelines in Section 1.4, the PPF is crucial in defining the set of storms to sample from. The set of storms must be determined before any probability distribution can be assigned to the i_r-t_r rainstorm combinations. Again examining the storm plots in Figures 2-3 through 2-6, one can visually approximate PPF from the 15 year collection of rainstorms. A systematic way of finding the PPF is important for consistant hydrologic modeling results. Ideally each region's PPF could be put into the same functional form. When in the same functional form, the individual PPF's could possibly be

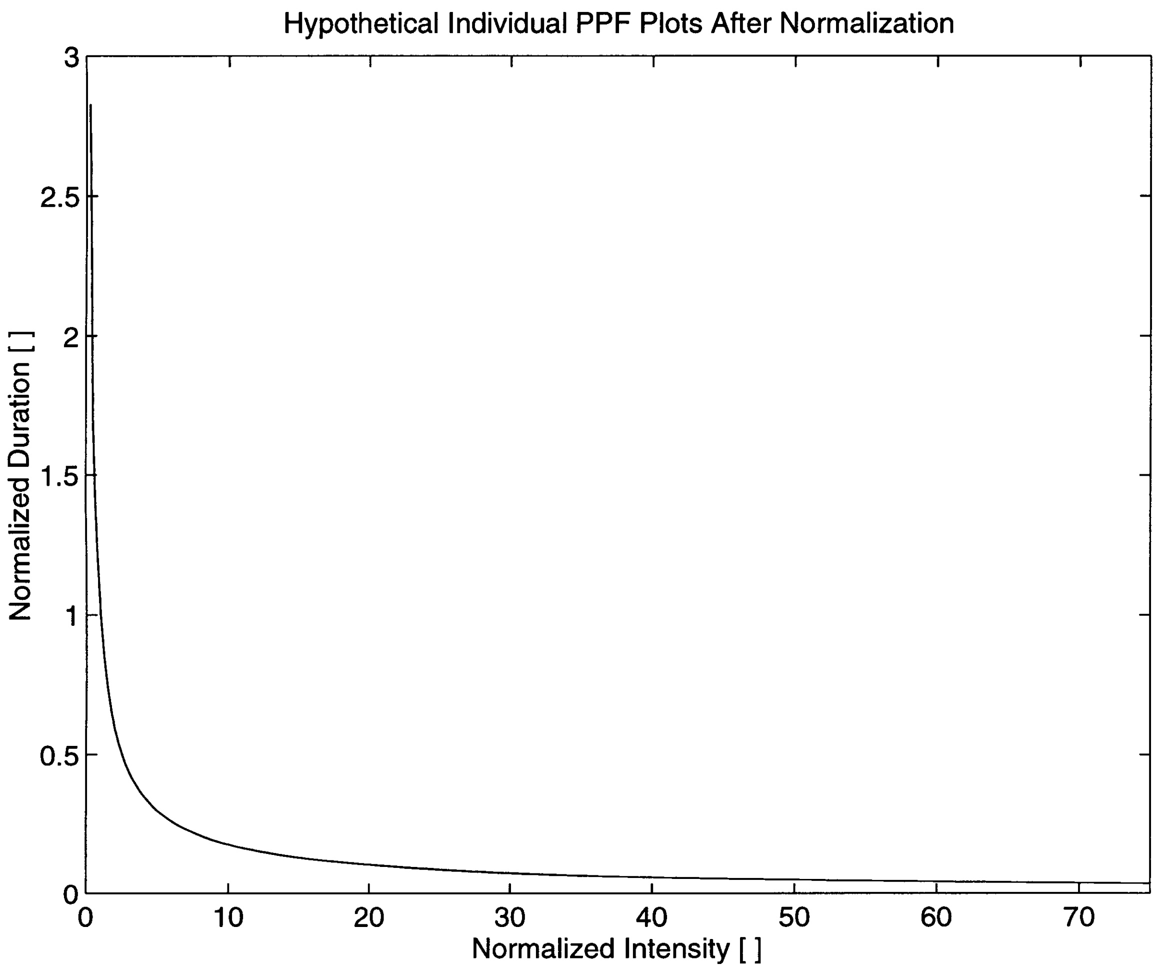

Figure 2-8: The individual PPFs collapse into a single universal non-dimensional curve upon normalization.

normalized by some readily observable parameters. That is, it is hoped that the individual PPFs will collapse into a single universal nondimensional curve (see sequential Figures 2-7 and 2-8) With the specified regional parameter values, it will be possible to recover a regional PPF without having to record a precipitation timeline. As a result, the PPF and feasible set of storms for a given region will be determined thereby creating sets of feasible storms (for sampling) for large regions whose preciptitation timelines have never been measured.

2.3.3 Selection of PPF Data Points: The GTBEFORE Algorithm

To evaluate various possible functional forms and normalizing parameters, the set of actual i_r-t_r data points, which are or on the PPF, must be found. For this purpose, once the set of storms has been found for a given region, an algorithm, developed for this study and known as GTBEFORE, is applied.[2] From the plot of storms, an axis is selected, say the i_r axis. Next, the i_r-t_r combination with the highest i_r is selected. Denote this combination as (i_{r_o}, t_{r_o}). This initial combination is one point on the PPF. The combination with the next highest i_r, (i_{r_1}, t_{r_1}) is then evaluated. If t_{r_1} is greater or equal to the t_{r_o}, then (i_{r_1},t_{r_1}) is a point on the PPF, if t_{r_1} is less than t_{r_o} then the combination is skipped. Each time a point is chosen for the PPF, it becomes the new (i_{r_o},t_{r_o}) used to evaluate the points with higher t_r. By this method, the point with the highest t_r is always chosen. That is, if there are two or more combinations with the same i_r, the combination that will be considered to be on the PPF will be the one with the highest t_r. This ensures that slope of the linearly interpolated PPF will be negative.

Next, the axes are interchanged and the process is repeated again. The final version of the PPF is the set of common combinations between the two runs of the GTBEFORE process. It has not been proved theoretically that an interchanging of

[2]The reader should refer to Appendix F for a detailed description and the computational algorithm.

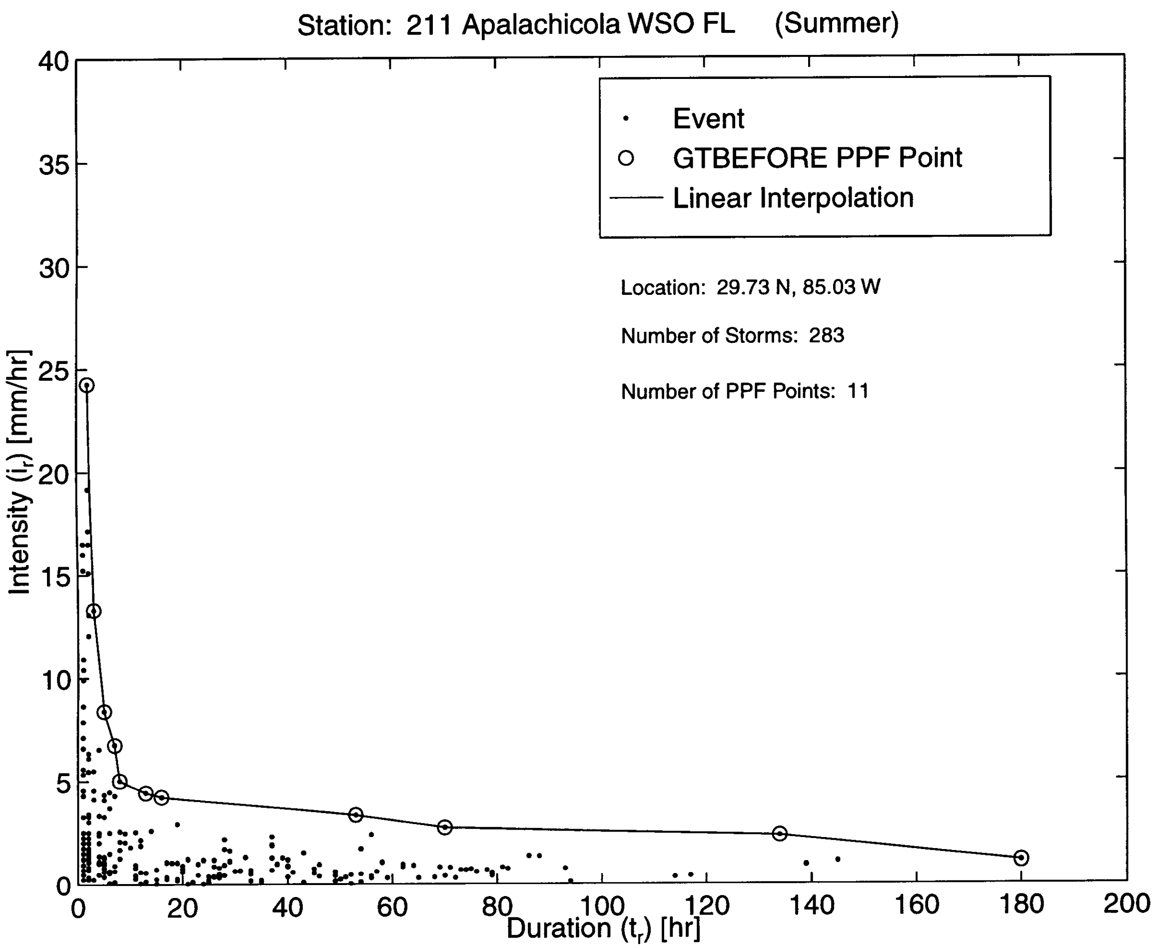

Figure 2-9: PPF data points chosen by the GTBEFORE algorithm (with linear interpolation) at Apalachicola, FL.

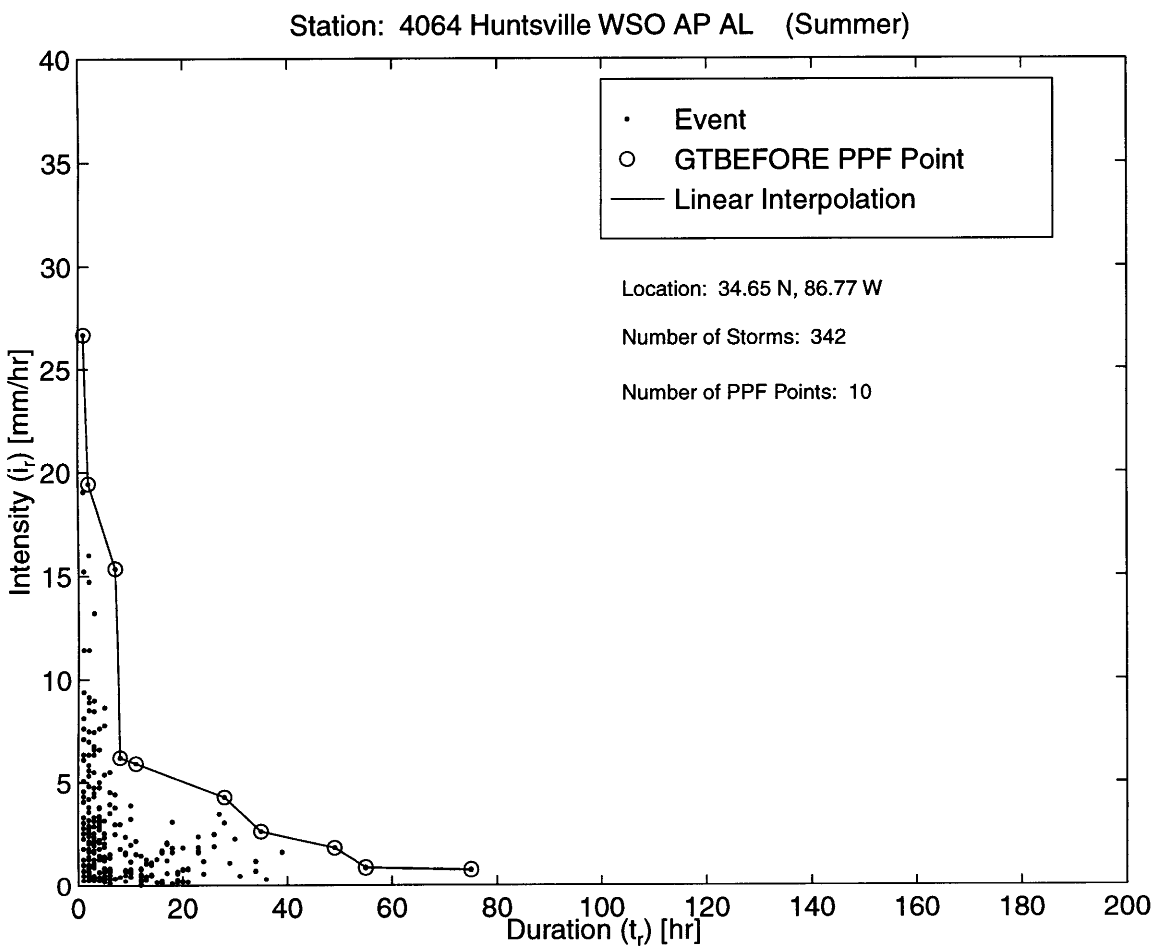

Figure 2-10: PPF data points chosen by the GTBEFORE algorithm (with linear interpolation) at Huntsville, AL.

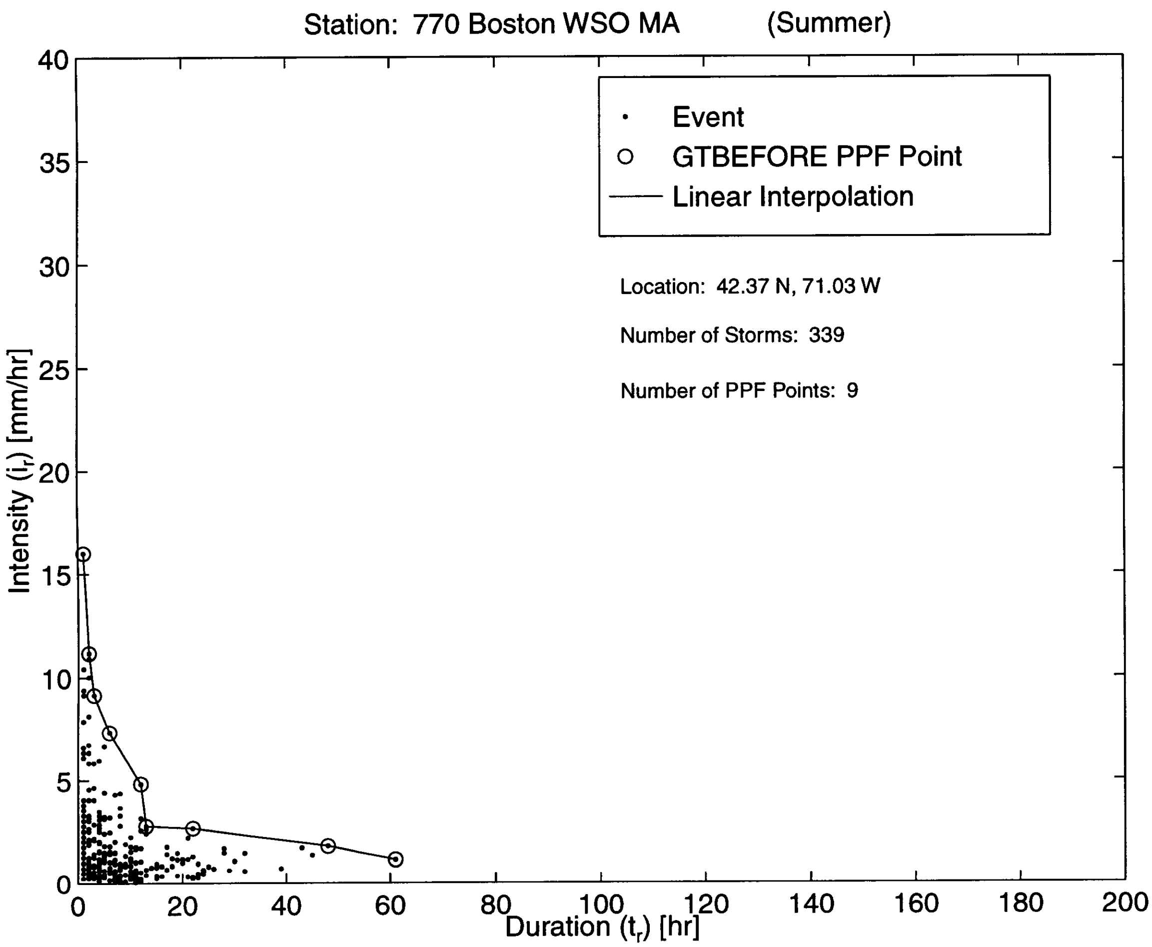

Figure 2-11: PPF data points chosen by the GTBEFORE algorithm (with linear interpolation) at Boston, MA.

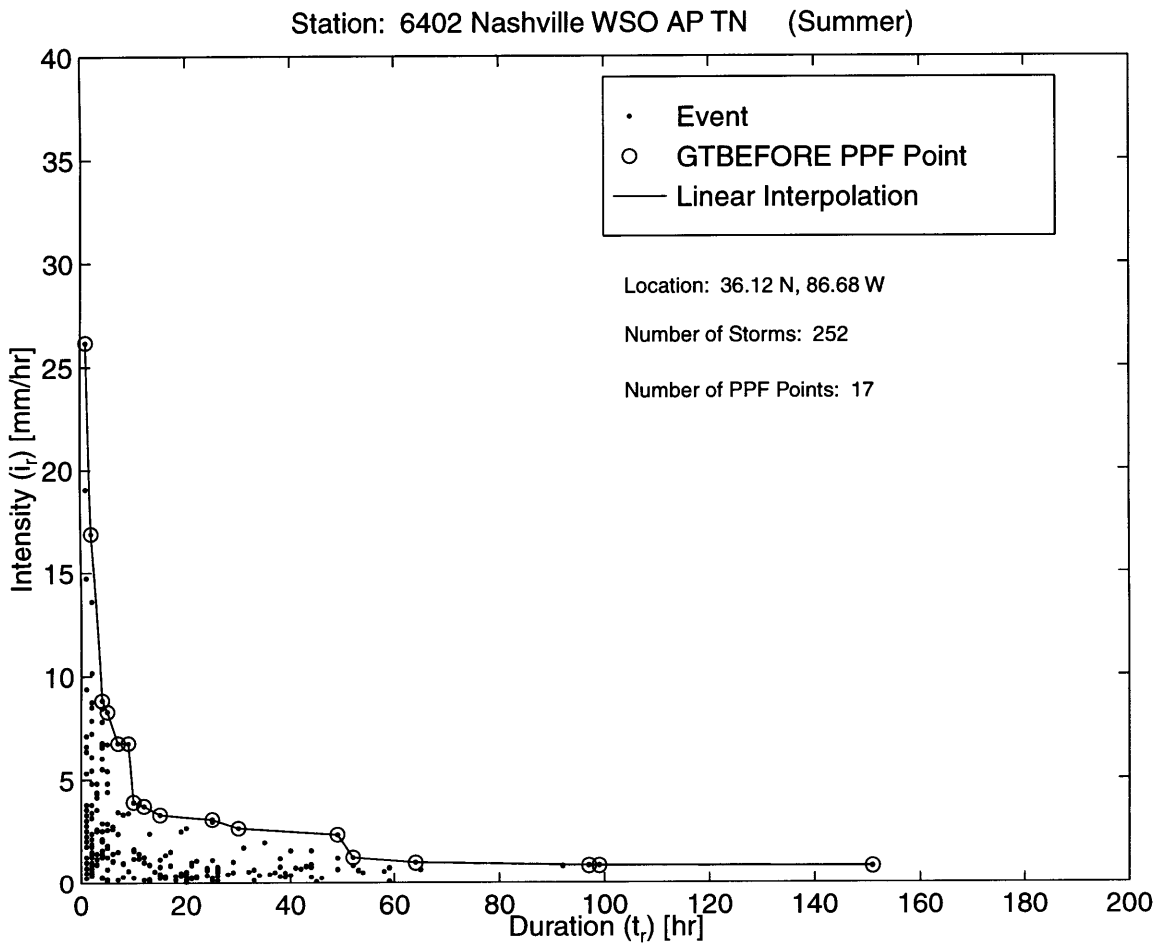

Figure 2-12: PPF data points chosen by the GTBEFORE algorithm (with linear interpolation) at Nashville, TN.

Rain Gauges in the U.S.

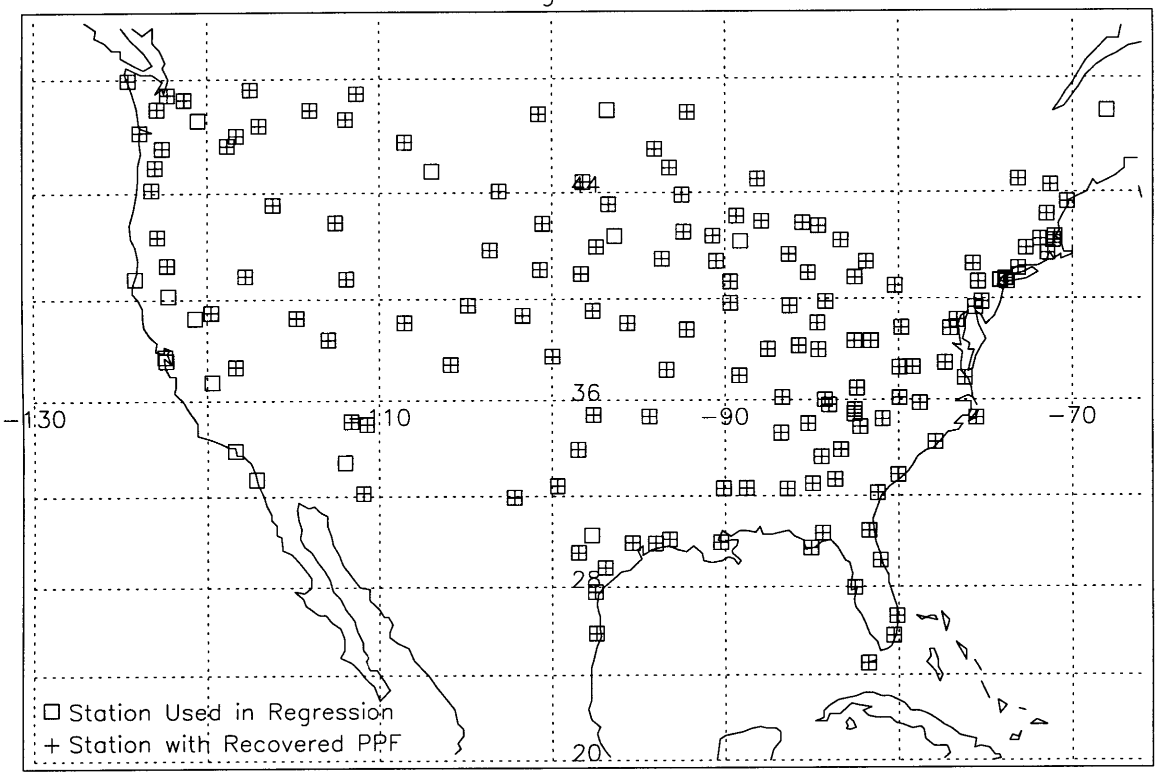

Figure 2-13: Rain gauge station locations and usage in analysis

the axes is a redundant process (i.e. yields the same points), but in tests of various weather station data, it does appear that an interchanging of the axes provides an identical PPF data set as that before the axes were interchanged. Figures 2-9 through 2-12 show the selection of PPF data points using the GTBEFORE algorithm on the example stations identified in Figures 2-3 through 2-6.

2.3.4 Station Choices for PPF Data Point Selection

Using EarthInfo [5] precipitation timelines from 1971 through 1985 for one hundred seventy four stations in the U.S.A., the same stations used by Wynn [33] in her analysis of the breakpoint T_{bmin}, we develop PPF curves for diverse regional climates. Wynn's values for T_{bmin} were used for the purposes of storm determination (see Appendix E). Of the 174 stations, only 161 stations were chosen because the other 13 stations had more than two days of data missing. Precipitation data was grouped into seasons: Spring (March–May), Summer (June–August), Autumn (September–November), and Winter (December–February). Seasons from specific years where the missing data was in that season were eliminated from the 15 year season count. Figure 2-13 shows the location and analysis usage of the 161 selected rain gauge stations.

2.4 The PPF Functional Form and Parameter Estimation

2.4.1 Functional Form

The most immediate and readily applicable method explored to create a PPF was to spline the points together through a cubic spline technique [17]. The benefits of this technique if applied successfully would be that the PPF would go through all the selected points and that a smoothness would be guaranteed. Points on the curve could be analysed statistically against the curve points of other stations to find if regional parameters do effect the curve generation. However, cubic splining proved to be far too unstable and the negative rate of transformation of the PPF could not be maintained.

Since the PPF's are generally hyperbolic in appearance, a basic functional fit equation of the form

$$\frac{i_r}{I} = C\left(\frac{t_r}{T}\right)^B, \tag{2.11}$$

where I and T are scale parameters, is selected. In order for units to match on both sides of the equality, I is in mm/hr, T is in hours, and C is a dimensionless constant.

Even though Equation 2.11 may not guarantee a perfect fit of the GTBEFORE points, it has some basic advantages. From the visual examination of the GTBEFORE plots of the various stations, it appears that required fitted curve would have to be negatively sloped and basically concave. The negative slope and downard characteristics of the GTBEFORE points can be seen in Figures 2-9 through 2-12. Because of the inclusion of I and T, the equation captures the possibility of normalizing parameters. If Equation 2.11 holds, then from one normalized curve, once the proper I and T for a given area are selected, the PPF for the area is known. Recalling Chapter 1, it is hoped that these I and T represent easily observed regional parameters. Another advantage of Equation 2.11 is that it is log-linear. Transforming Equation 2.11 by taking the natural log of both sides, the linear equation

$$\ln\left(\frac{i_r}{I}\right) = \ln C + B\ln\left(\frac{t_r}{T}\right). \tag{2.12}$$

results.

Because Equation 2.12 is linear and overdetermined, one can apply an ordinary least squares regression (OLS) upon the model for parameter estimation.

$$\ln\left(\frac{i_{r(jn)}}{I_n}\right) = A + B\ln\left(\frac{t_{r(jn)}}{T_n}\right) + \varepsilon_{jn} \tag{2.13}$$

where n denotes the n^{th} GTBEFORE combination of station j, and I_n and T_n are the I and T parameter values for station n.

We assume the model specified by Equation 2.13 to be correct. There is no specific reason as to why the regression could not apply in reverse. That is, intead we regress $\ln\left(\frac{t_{r(jn)}}{T_n}\right)$ upon $\ln\left(\frac{i_{r(jn)}}{I_n}\right)$. This is tantamount to assuming that the basic equation is

$$\frac{t_r}{T} = Z\left(\frac{i_r}{I}\right)^F \tag{2.14}$$

whose natural log linear transformation would be

$$\ln\left(\frac{t_r}{T}\right) = \exp(M) + D\ln\left(\frac{i_r}{I}\right) \tag{2.15}$$

where $\exp(M) = Z$. Unless the R^2 value of the regression of Equation 2.13 is exactly one, the coefficients for the reverse regression variable will not be an exact inverse. B of Equation 2.13 will be related to a coefficient D of the reverse regression

(Equation 2.14) [30] by

$$B = \frac{(corr[\ln\left(\frac{t_{r(jn)}}{T_n}\right), \ln\left(\frac{i_{r(jn)}}{I_n}\right)])^2}{D} \tag{2.16}$$

Such an examination of the effect of a reverse regression is beyond the scope of this thesis.

2.4.2 The Gauss-Markov Assumptions

In order to successfully apply ordinary least squares, the Gauss Markov assumptions which are critical in ensuring that the regression coefficients A and B are $BLUE$[3], must be checked [22]. We begin with a discussion of the least important assumptions first and proceed to the most important assumptions.

1. ε_{jn} is normally distributed. Since the the data set is large for the regression (161 stations providing the GTBEFORE points), then one can assume asymptotic normality of the error term ε_{jn}. This assumption allows statistical tests.

2. No autocorrelation of the error term. Another statement of this assumption is that the variance-covariance matrix of the errors ε is $\sigma^2 I$ where I is an identity matrix [22]. Because the regression data represents the gathering of the data from the 161 stations, serial correlation for the data set as a whole seems unlikely. However within an individual an individual station's GTBEFORE series of points, serial correlation of the error term may exist. Heteroskedasiticity, however, may exist. In heteroskedasticity, the error variance is a function of the explanatory variables. In a bivariate regression (constant and variable), heteroskedasticity can sometimes be detected by visual examination of the change of the data points spread around the various regression lines along the variable axis.

To test for heteroskedasticity, a White test on the residual error term is carried

[3] *B*est *L*inear *U*nbiased *E*stimator

out. An OLS regression is run upon the residual error term where the regression equation is specified as

$$\hat{\varepsilon_{jn}}^2 = \nu + \gamma_1 \ln\left(\frac{t_{r(jn)}}{T_n}\right) + \gamma_2 \left(\ln\left(\frac{t_{r(jn)}}{T_n}\right)\right)^2 + \delta_{jn} \tag{2.17}$$

where ν is a regression constant and δ_{jn} can be interpreted as the error term of the White test regression. The inclusion of the variable $\left(\ln \frac{t_{r(jn)}}{T_n}\right)^2$ reflects the possibility that ε may have a non-linear relation ship to $\ln\left(\frac{t_{r(jn)}}{T_n}\right)$ to reflect possibilities of the relation of the error term variance to linear and non-linear forms of $\ln\left(\frac{t_{r(jn)}}{T_n}\right)$. The White statistic

$$NR^2 \sim \chi^2_2 \tag{2.18}$$

where N is the total number of data points and R^2 is the R^2 statistic of the OLS regression of Equation 2.17, tests the null hypothesis $h_\circ$ of homoskedasticity. If the null is rejected (at some arbitrary level of significance) and heteroskedasticity is thought to exist, the standard errors of the coefficients may be off and the test statistics may be invalid. Since there is a large number of data points, it is hoped that the regression upon Equation 2.13 asymptotically homes in upon the true values of the coefficients A and B, or at least comes to within a tolerable margin from the true values. Such a large sample property is known as consistancy (see assumption 3). A correction scheme for heteroskedasticity was not conducted because of the complexity of the correction analysis (White correction). Instead, the asymptotic consistancy property assumption was relied upon in providing the correct values for A and B. However, an indication of autocorrelation at the large scale, could be important, especially if the variance of the error terms does not shrink as the sample size grows. Heteroskedasticity does definetly imply that the standard errors are incorrect.

3. $\mathbf{E}(\varepsilon_{jn} | \ln\left(\frac{t_{r(jn)}}{T_n}\right) = 0$. If this assumption is not true then the coefficient estimates are both biased and inefficient. Nor is there any guarantee that the estimates will be consistant. Violations of this assumption are hard to detect. It is a very important assumption and hard to correct for if it does not hold. If the assumption fails, another variable must be found correlated $\ln\left(\frac{t_{r(jn)}}{T_n}\right)$ but asymptotically uncorrelated to the error term and a technique called instrumental variables two stage least squares must be conducted. This is complex and for the purposes of this thesis, it is assumed that the error term is uncorrelated with the regression variables.

4. Existance. The inverse of the regression matrix exists otherwise the regression procedure would fail. For all parameter combinations the ordinary least squares technique succeeds.

Should OLS provide viable estimates of A and B, then to find a recovered station PPF it is a simple matter to apply the coefficients to Equation 2.11 and for, a given station, find the stations I and T values.

Chapter 3

Modeling and Results

3.1 Stations and Station Statistics

Appendix A lists the 174 raingauge stations and their corresponding latitude and longitude coordinates. Recall that in Subsection 2.3.4 only 161 stations were used due to missing records at the 13 eliminated stations. Figure 2-13 reveals the diversity of climate regimes that the 161 stations are located in. The climates vary from the more arid and desert-like conditions of the Southwest to the more tropical areas of Florida to the more temperate climates of New England and colder regions of the Northern mainland United States. The summer season, which is chosen for the analysis, offers several advantages. First of all, for the encompassed climates, the summer precipitation is in the form of of rain. This avoids the problems that raingauges suffer when measuring snowfall (see Subsection 1.3.1). Secondly, the rainstorms of the summer are more convective and localized rather than large scale synoptic system driven.

The GTBEFORE PPFs of 144 of the 161 stations are overplotted on Figure 3-1. The 144 stations are marked with a + within the □ sign in Figure 2-13. The seventeen unplotted stations are eliminated because they each fail to have more than six GTBEFORE PPF points. It is considered that fewer than seven points cannot give a good characterization of an individual PPF. These left out points, however, may still be on a PPF though and are included in the regression analysis. Many of the 17 eliminated stations are in regions where it is very dry in the summer. In such

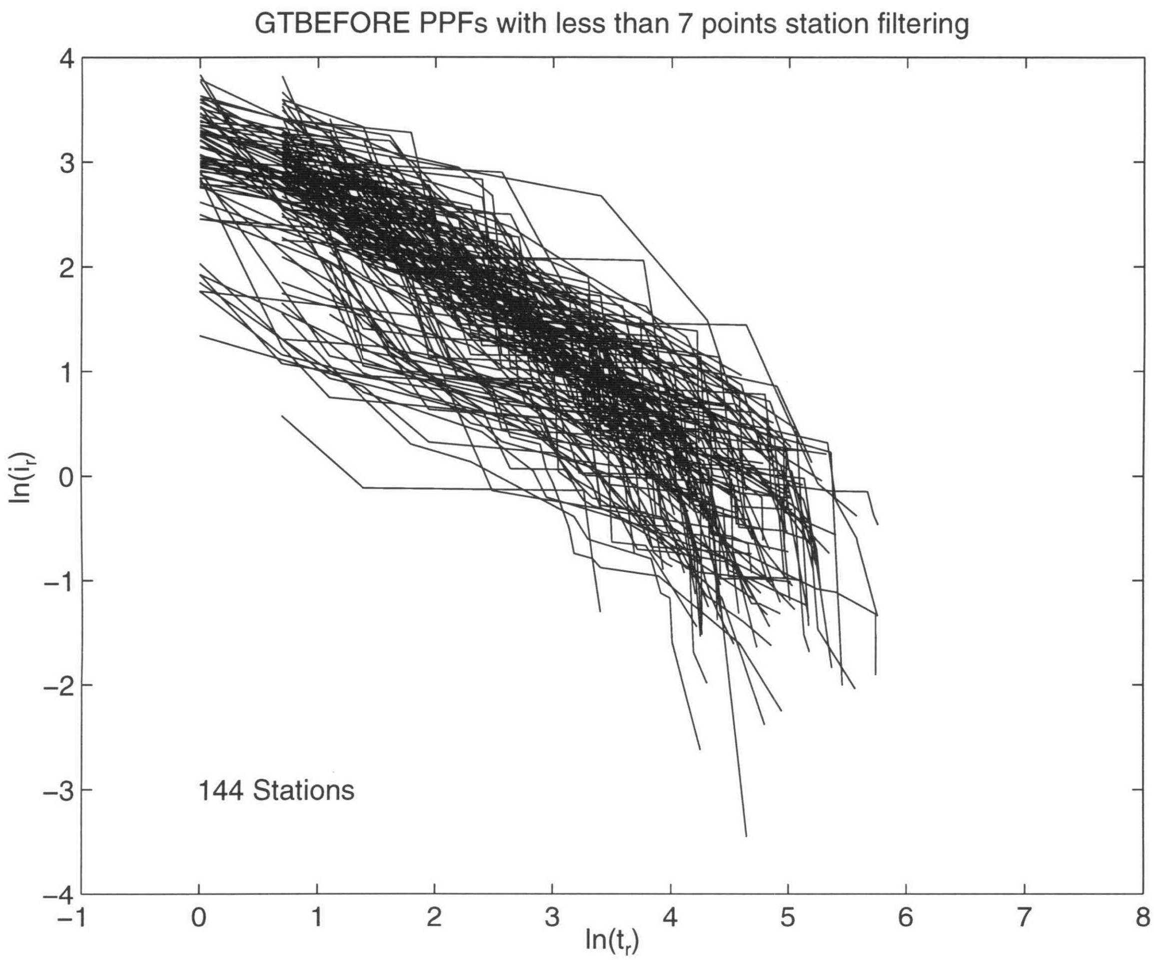

Figure 3-1: Natural log-log plots of the GTBEFORE PPF curves from 144 raingauge stations.

places (e.g. California), one notices that the values of the Rectangular Pulse Model $E(t_b)$ in Table B.2 are unusually high. To illustrate with an example, when $E(t_b)$ is greater or equal to 168 hours for a season, there are less than fourteen rain pulses on average for the season of each year.[1] It follows that the characterizing of the PPFs of arid regions is crippled from the lack of a sufficient number of storms.

Figure 3-1 reveals a basic natural log-log linearity relation between the PPF intensities and durations. This seems to support the notion that Equation 2.11,

$$\frac{i_r}{I} = C\left(\frac{t_r}{T}\right)^B, \tag{3.1}$$

can be a proper specification because it too is natural log-log linear. This is seen by its transformation into Equation 2.12,

$$\ln\left(\frac{i_r}{I}\right) = \ln C + B\ln\left(\frac{t_r}{T}\right), \tag{3.2}$$

by taking the natural log of both sides of Equation 2.11. We refer to $\ln C$ as the constant A for notational simplicity. B is interpreted as a measurement of the elasticity of i_r/I with respect to t_r/T. That is, letting $G = i_r/I$ and $H = t_r/T$,

$$e_{G,H} = \frac{\Delta\%\ in\ G}{\Delta\%\ in\ H} = \frac{\Delta G/G}{\Delta H/H} = \left(\frac{dG}{dH}\right)\frac{H}{G} = B. \tag{3.3}$$

Thus B indicates how G responds to a one percent increase in H, *ceteris paribus*[21]. From the form of Equation 2.11, B is held to a constant.

Another important observation from Figure 3-1 is that the natural log-linear stations PPFs cover a wide range of scales (similar to Figure 2-7). If regional parameters are found to non-dimensionalize the station PPFs and to reduce the scatter of the curves (as in Figure 2-8), then a universal PPF may be defined.

[1]Note that $E(t_b)$ is derived from the Rectangular Pulses Model which has its own assumptions about the distributions of i_r and t_r. Over long periods of time, $E(t_b)$ and the other RPM statistics in Table B.2, do approach the sample values.

3.2 Parameter Values of I and T

In reviewing Figures 2-7 and 2-8 in Chapter 2, it is put forth that Equation 2.11 represents the universal normalized PPF from which all of the regional PPFs can be found. Therefore a universal regression is run upon Equation 2.13, the natural log-linear transformation of Equation 2.11, where the endogenous and exongenous variables are the 1516 i_r-t_r combinations selected from the 161 stations by the GT-BEFORE alogorithm.

The essential question is whether or not the choices of I and T (the normalizing or scale parameters) show some regional characteristics so that the universal functional form of the PPF, Equation 2.11, may be used to define the envelope of feasible i_r and t_r values.

Since I and T have dimensions [L/T] and [T] respectively, we select a number of dimensionally the same variables with regional behavior (i.e. they are related to general hydrometeorological conditions) for the purposes of relating them to I and T. Examples of these regional hydrometeorological variables are given in Table 3.2. The mean seasonal precipitation rate (μ), the conditional mean precipitation rate conditioned on rain event ($\mu/Pr(0)$), standard deviation of rain rate σ, and the RPM mean storm intensity parameter (E(i_r)) are all characterized by [L/T] dimensions. These form suitable candidate variables for developing a predictive and regional model for I in Equation 2.11. For the T parameter with time dimension, the RPM parameters for mean storm duration (E(t_r)) and mean interstorm arrival (E(t_b)) are viable candidates. Similarly, the T_{bmin} parameter may be used. Finally the precipitable water W (integral of water vapor in the air column represented as equivalent depth of liquid water) divided by the mean precipitation rate forms a time-scale for the turn-over of the atmosphere. This regional variable will also be used to model T.

With the set of I and T parameters determined for each station (Appendix B), the normalized axes for i_r and t_r can be defined. Figure 3-2 is an example showing the normalization of the intensity axis by the mean seasonal precipitation rate (μ), and the normalization of the storm duration axis by the atmospheric vapor turn-over

Table 3.1: Variables for Parameters I and T

I [mm/hr]	T [hr]
μ	W/μ
$\mu/Pr(0)$	$E(t_r)$
σ	$E(t_b)$
$E(i_r)$	$T_{bmin(BP)}$

time-scale (W/μ). Appendix C contains the combination of normalizing axes figures similar to Figure 3-2 but for the other scale parameters in Table 3.2.

The remaining task is to find the normalizing regional coefficients that form the least scatter around the log-log linear model for the PPF. The universal PPF would be a straight line in log-log normalized storm intensity and duration space. This straight line is determined by the OLS regression described at the head of this section (Section 3.2). The normalizing scale parameters that account the best for regionalization would create the least amount of scatter around the universal PPF.

Statistics and measures that quantify the scatter need to be defined. The explained variance (R^2) is clearly an applicable criteria. Nevertheless since the axes are natural log-transformed, statistical optimality is not guaranteed with this statistic. Visual goodness of fit is also employed to find the normalizing set of variables that create the least amount of scatter about the universal PPF.

The parameter combinations in Figure 3-2 fulfill the selection requirements described in the previous paragraph. Note that the R^2 statistic is the highest and the normalization parameters μ and W/μ give the tightest linear compression on the log-log graph. The White test statistic shows that there is evidence of heteroskedasticity Nevertheless, the extremely high t-test (and very low standard error) values imply that that the coefficients for A and B are significant.[2] Another observation that reinforces this model parameters choice is that the sign unrestricted estimate of B is negative. This fits with assumption that the PPF is both concave and that the rate

[2]A more detailed description of the statistical tests is in Appendix C, Section C.3.

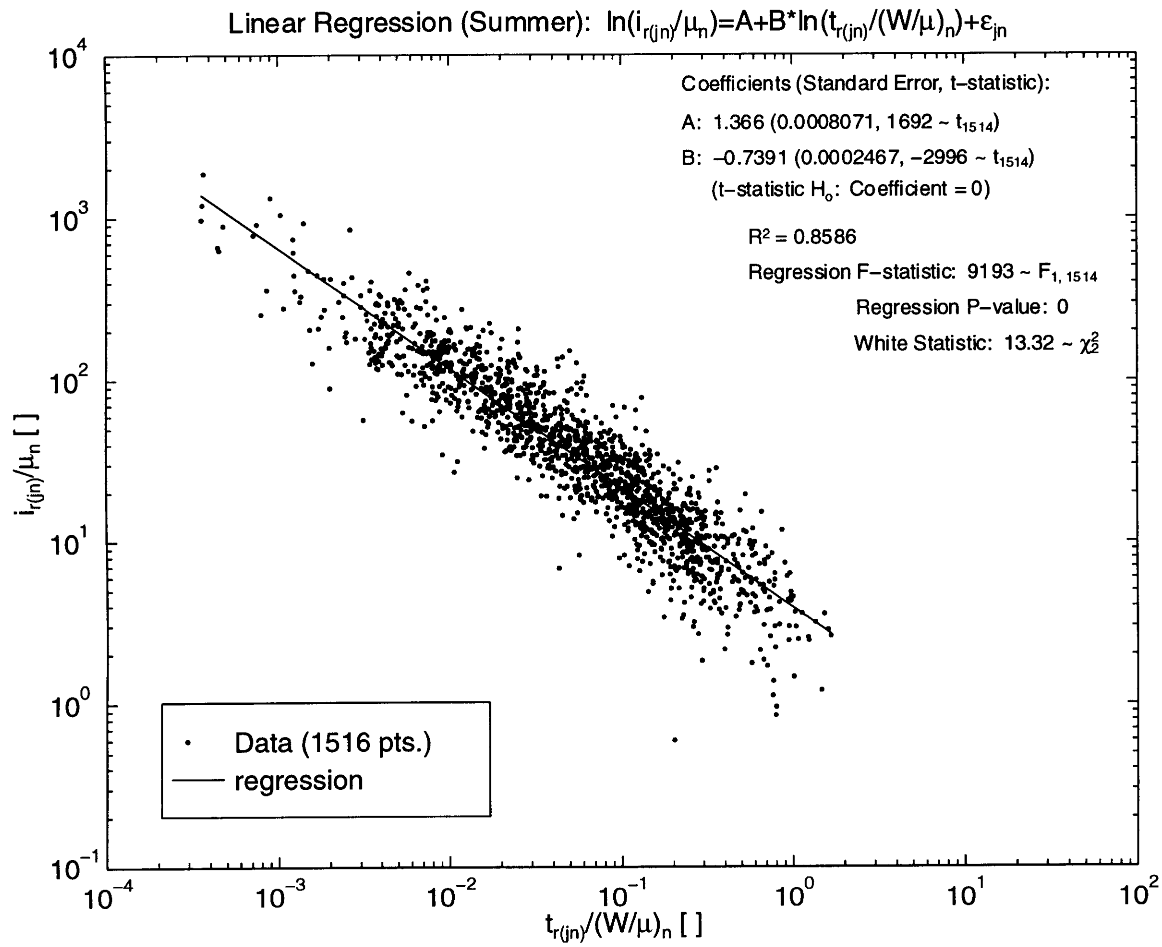

Figure 3-2: Ordinary Least Squares Regression on Equation 2.13 with $I = \mu$ and $T = W/\mu$. The figure is also in Appendix C.

of transformation (or slope) is indeed negative.

From an *a posteriori* view, the choice of $I = \mu$ and $T = W/\mu$ make sense. μ, the sample mean precipitation reflects the average rate of rainfall. Likewise with W/μ representing the average residence time of a particle of water in the atmosphere (see Appendix B), a higher W/μ may imply higher durations of storms in general. Because storms may be of higher duration and intensity if I and T are higher, the PPF may be spaced more outward (positive X and Y direction). Hopefully, I and T, when used to normalize the intensities and durations, collapse the individual PPFs into one general nondimensional curve.

3.3 Station Analysis

3.3.1 Standard Statistics and Problems

Once the specific I and T parameter model is chosen, the functional form PPF can now be recovered. The individual station parameter values of I and T (μ and W/μ respectively), along with the model coefficients (where $C = \exp(A)$), are entered into Equation 2.11. For the four example stations, the equation is plotted against the linearly interpolated GTBEFORE selected PPF.

Since the axes of the graphs are in different dimensions, statistical evaluation of the goodness of fit of the curves is difficult and it is hard to make independent of units. The primary evaluation must be visual. It can be seen that the general shape of the recovered curves for the four stations follow the pattern of the linearly interpolated curves formed by the data points. Note that the recovered PPF of Nashville, TN (Figure 3-6) fits rather well despite having seventeen points. The statistical computations shown in Figures 3-3 through 3-6 for evaluating the effectiveness of Equation 2.11 are described in Appendix D. In an attempt to visually compensate for the apparent large errors at the extremes introduced by the hyperbolic nature of the curves, log-log transformations of the recovered PPF and GTBEFORE PPFs of Figures 3-3 through 3-6, are shown in Figures D-1- D-4.

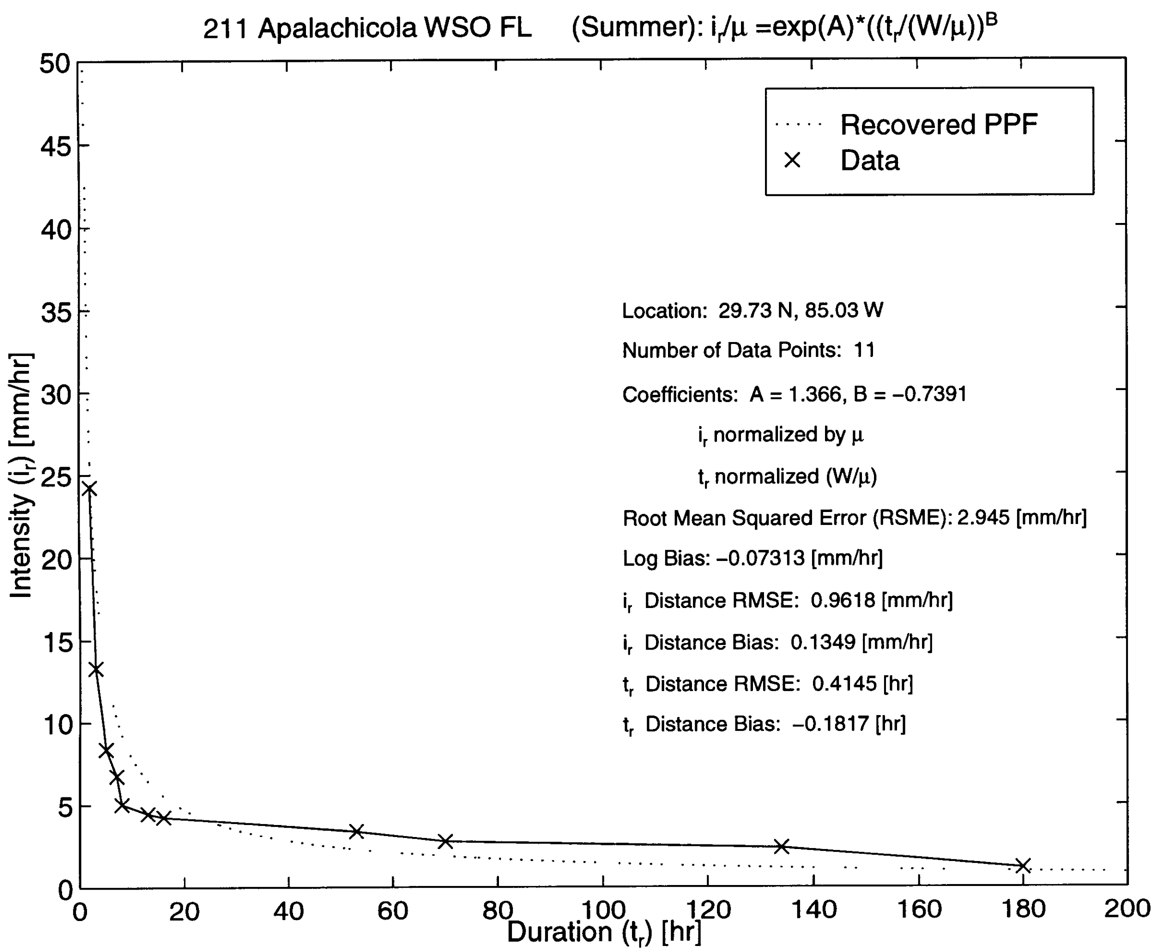

Figure 3-3: Recovered PPF, actual (GTBEFORE) PPF, and comparative statistics for Apalachicola, FL.

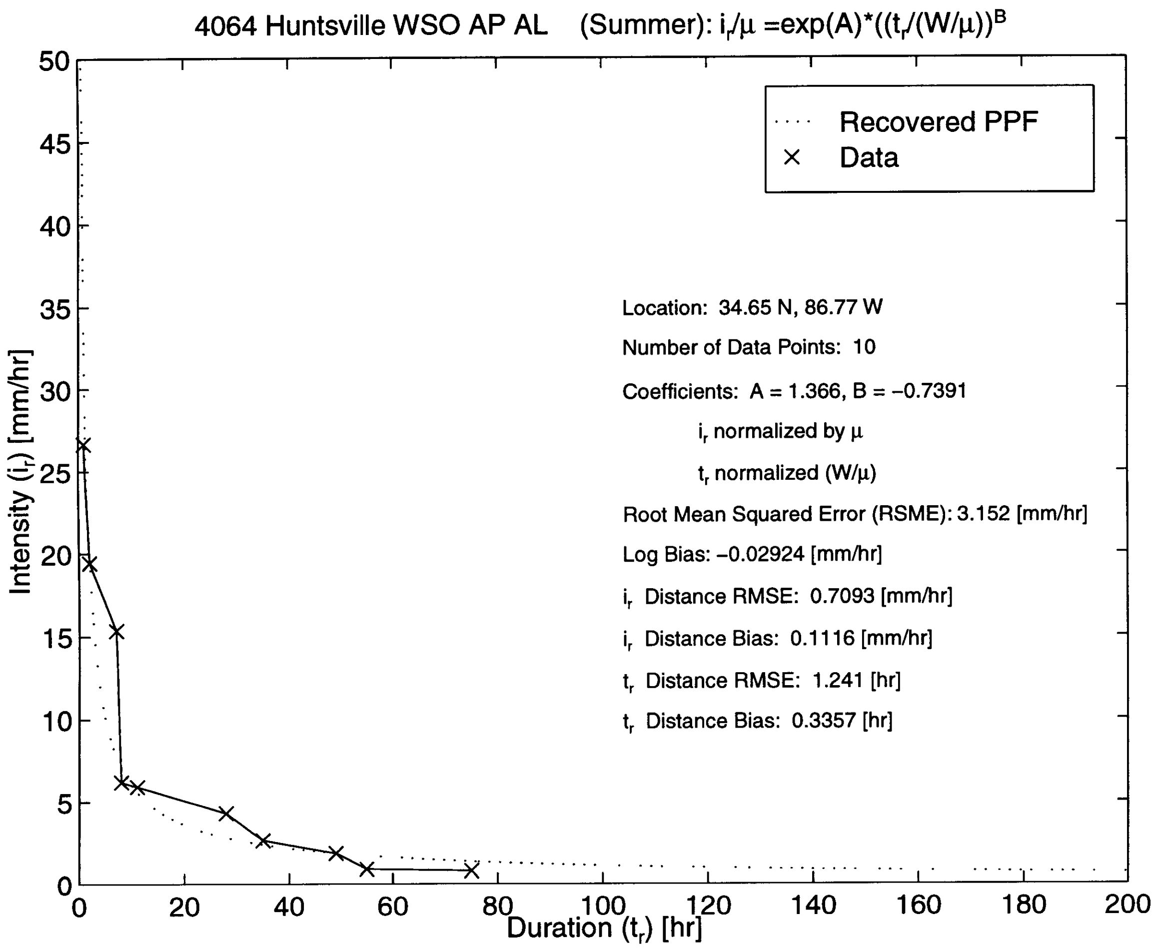

Figure 3-4: Recovered PPF, actual (GTBEFORE) PPF, and comparative statistics for Huntsville, AL.

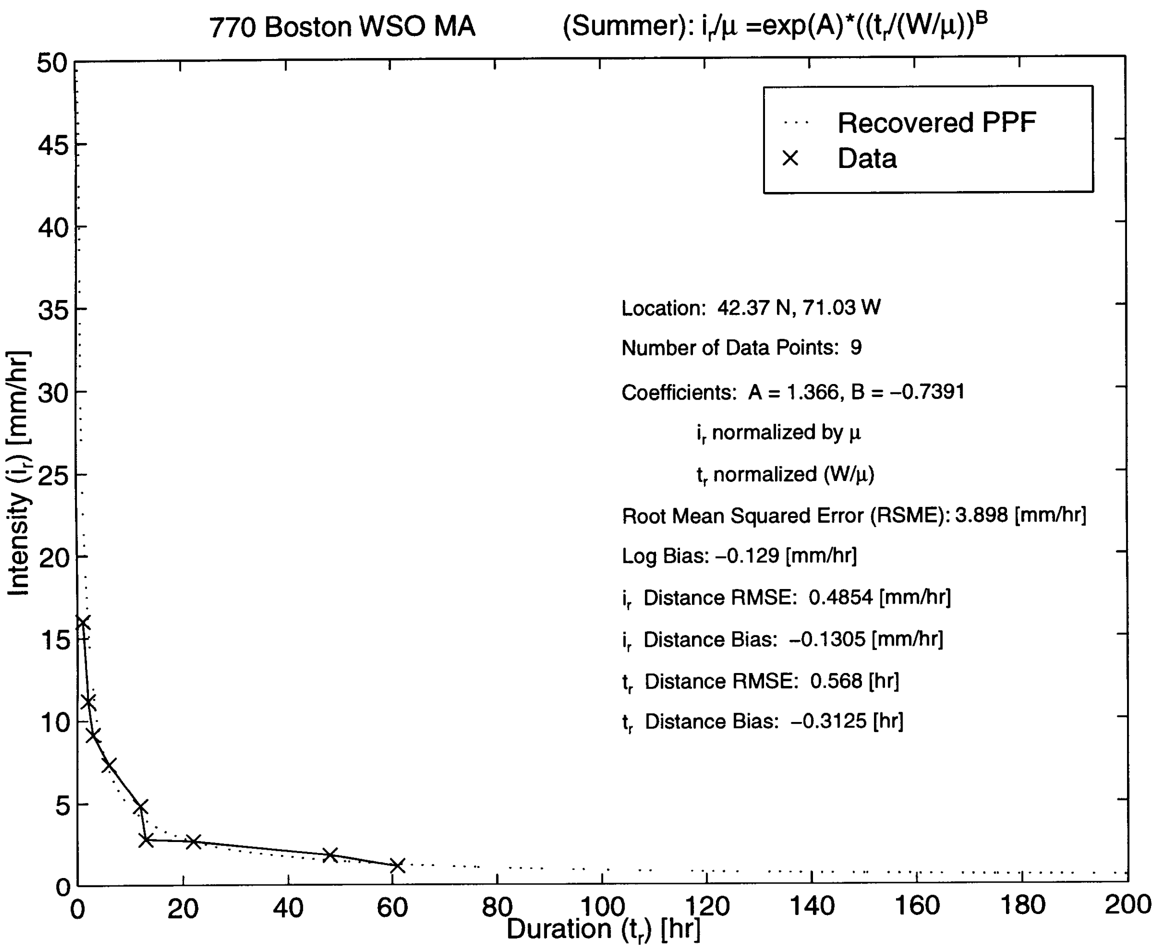

Figure 3-5: Recovered PPF, actual (GTBEFORE) PPF, and comparative statistics for Boston, MA.

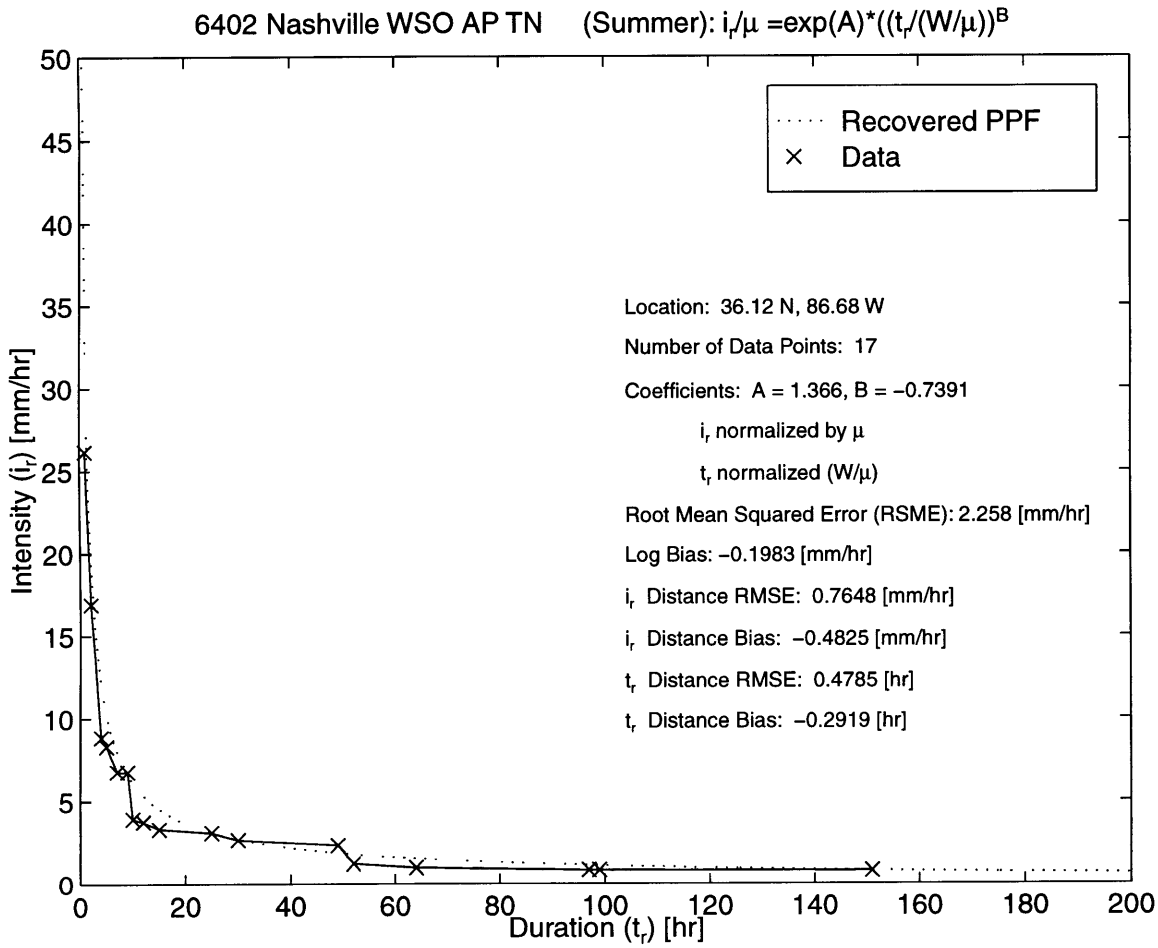

Figure 3-6: Recovered PPF, actual (GTBEFORE) PPF, and comparative statistics for Nashville, TN.

Table 3.2: Individual station coefficients summary when $I = \mu$ and $T = W/\mu$.

Description	$A(=\ln C)$	B	C
Minimum	-0.9342	-1.4647	0.3929
Maximum	2.7573	-0.4275	15.7576
Sample Mean	1.2422	-0.7886	3.9255
Variance	0.2822	0.0281	4.0943
General Model	1.3657	-0.7391	3.9185

One final evaluation method is to run the regression of Equation 2.13 upon individual stations with the I and T parameters of the chosen model above. This enables the comparison of individual stations against the general combined stations model by examining the spread of the individual station coefficients A and B upon the general model coefficients. The regression of the individual stations allows for station uniquenesses that are uncaptured by I and T to be accounted for. For $I = \mu$ and $T = W/\mu$ the coefficients for A, B, and C are listed in Appendix D.[3] A summary of the individual station coefficients is given below in Table 3.3.1. Again note that the 144 stations used here passed the criteria of having seven or more GTBEFORE selected points:

The sample mean and variance are not exactly the true estimators for the station choices is not random. Nevertheless notice that the means are not too far away from the general model values and that the signs agree. Of course C, because it is the exponential transformation of A, exhibits more of a spread. Another critical factor is that the minimum and maximum values of B agree in sign with the general model. This shows that the rate of transformation of the individual PPFs is indeed negative. The average idividual and general model values of B are between -1 and 0. This indicates a general inelasticity i_r/μ with respect to $t_r/(W/\mu)$.

With the individual coefficients, t-tests can be conducted in the same way as discussed in Appendix C, Equation C.4 where the null hypothesis values are the

[3]From Equation 2.12, C is $\exp(A)$.

coefficient values of the general model.[4] This enables us to test, through t-distribution analysis, if the individual station coefficients values are statistically different than the general model values for a chosen level of significance.[5] These t-statistics are listed for the various stations in Table D.3. The number of points per station is included in Tables D.2 and D.3. Again note that the number of explanatory variables in the regression k (including the constant) is 2. The points per station and k are needed for computing the degrees of freedom in the t-tests.

Figures 3-7 and 3-8 show the distribution of the individual station values for C and B, while contour plots are shown in Figures 3-9 and 3-10.

Visually, on the example stations in this thesis, the general and individual models give PPFs quite close to each other as can be seen in Figures 3-11 through 3-14.

3.3.2 Compression and Conclusions

To conclude the results we show Figure 3-15. In this figure, the number of log cycles on the X and Y axes has been kept the same as in Figure 3-1. Notice that the normalization of i_r and t_r has compressed the the plot of Figure 3-1 into a more linear form—from an approximate spread of three log cycles to two. This is the general result that we seek. Tighter the compression, the closer Equation 2.11, with the normalizing station parameters and proper coefficients comes to approximating the GTBEFORE PPFs.

The selection of μ and W is convenient. Both can be observed on satellite platforms (μ on a monthly basis) worldwide. The model with the selected parameters retains a simplicity desired for engineering purposes. This work demonstrates that to recover the PPF of an unknown, unmeasured, non-raingauged area, one may just have to know know the mean precipitation and precipitable water of the area along

[4]Since the individual station regressions are done on small samples (i.e. low numbers of GTBEFORE points) there is a stronger possibility that heteroskedasticity and/or serial correlation may have more influence on the efficiency of the estimators. Because of this, White test statistics on Equation 2.17 for individual stations, and Durbin-Watson statistics (for first degree serial correlation) have been included in Table D.1. As stated in Appendix D, efficiency correction is beyond the scope of this work. The reader can consult *Pindyck and Rubinfeld* [22] for more details.

[5]The desired level of significance is left to the reader to decide.

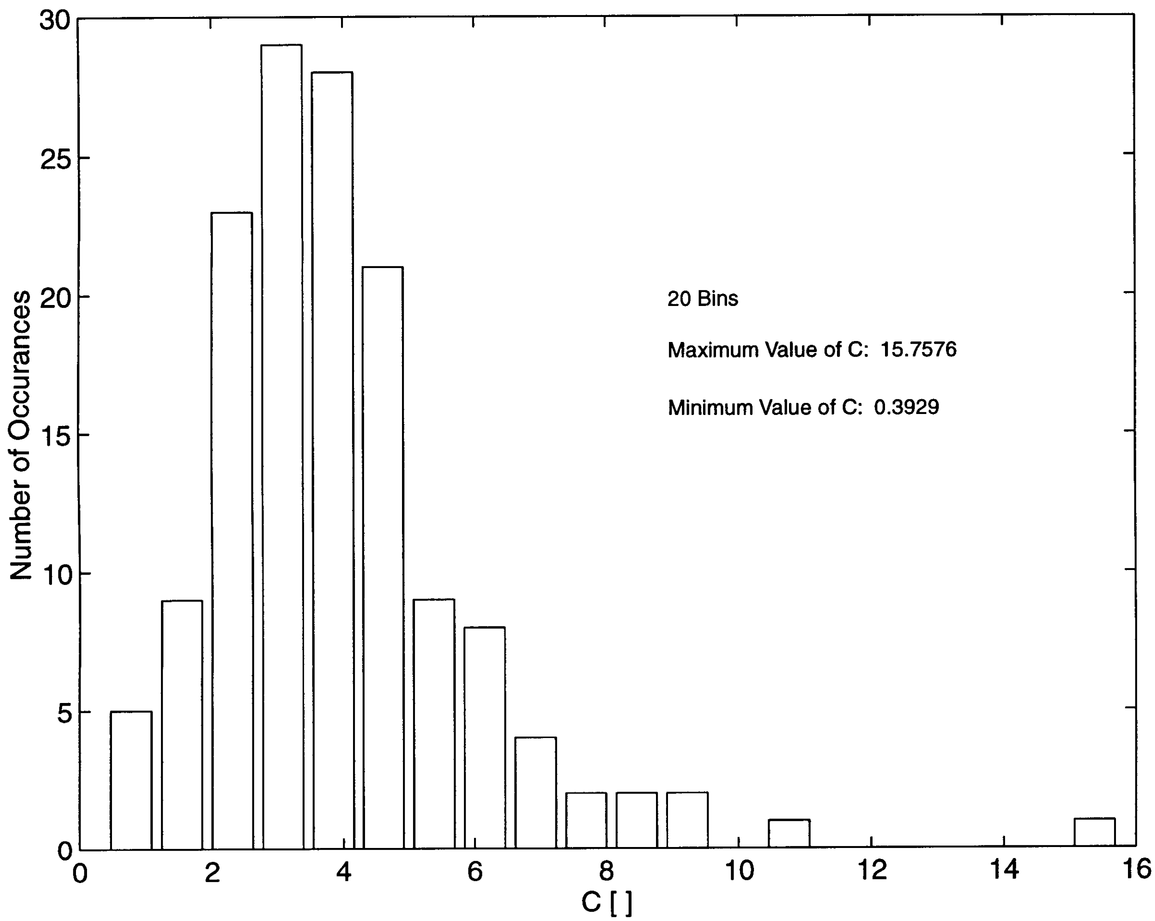

Figure 3-7: Histogram of the individual station coefficient values of C.

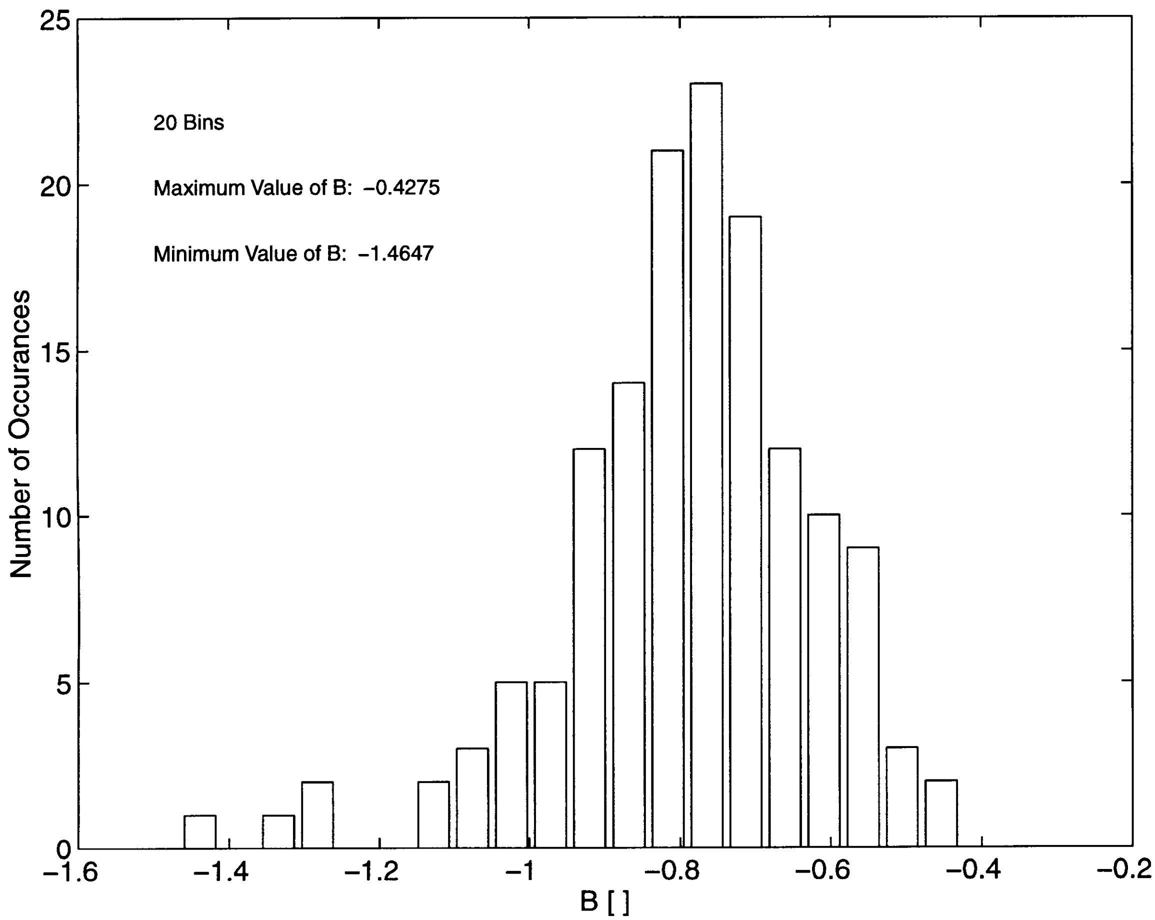

Figure 3-8: Histogram of the individual station coefficient values of B.

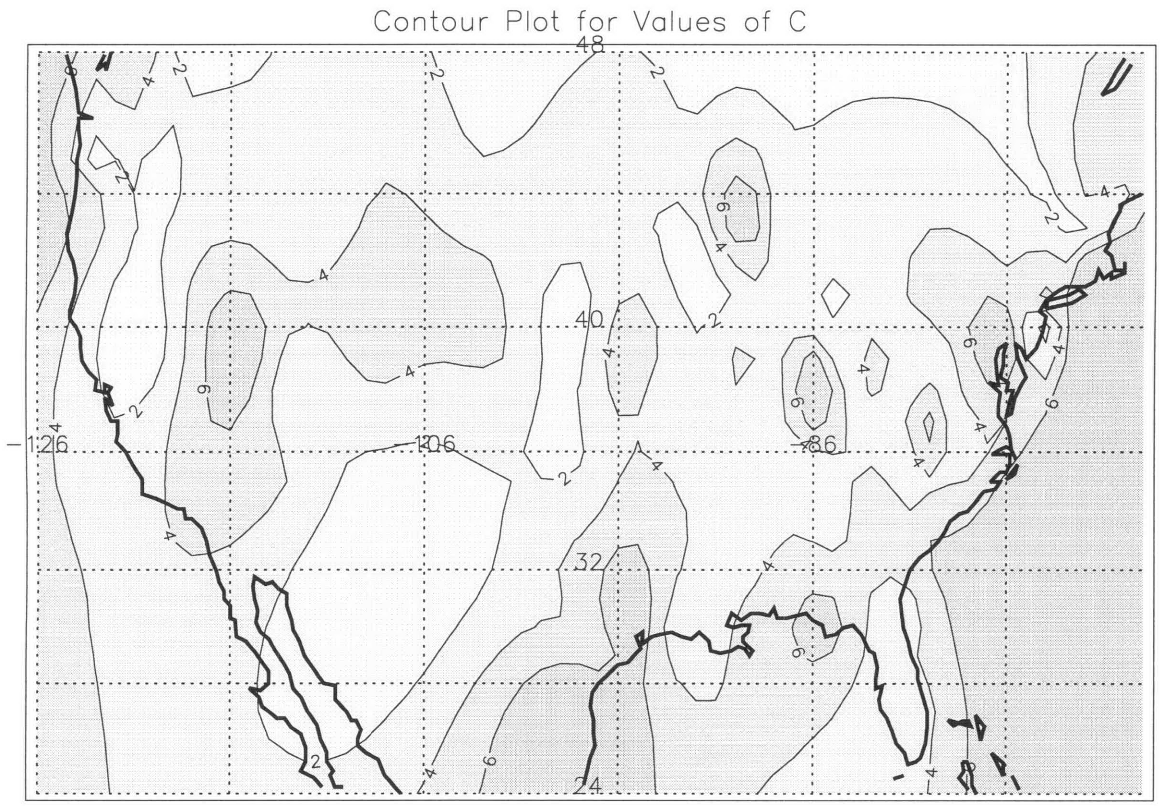

Figure 3-9: Contour plot of the individual station coefficient values of C.

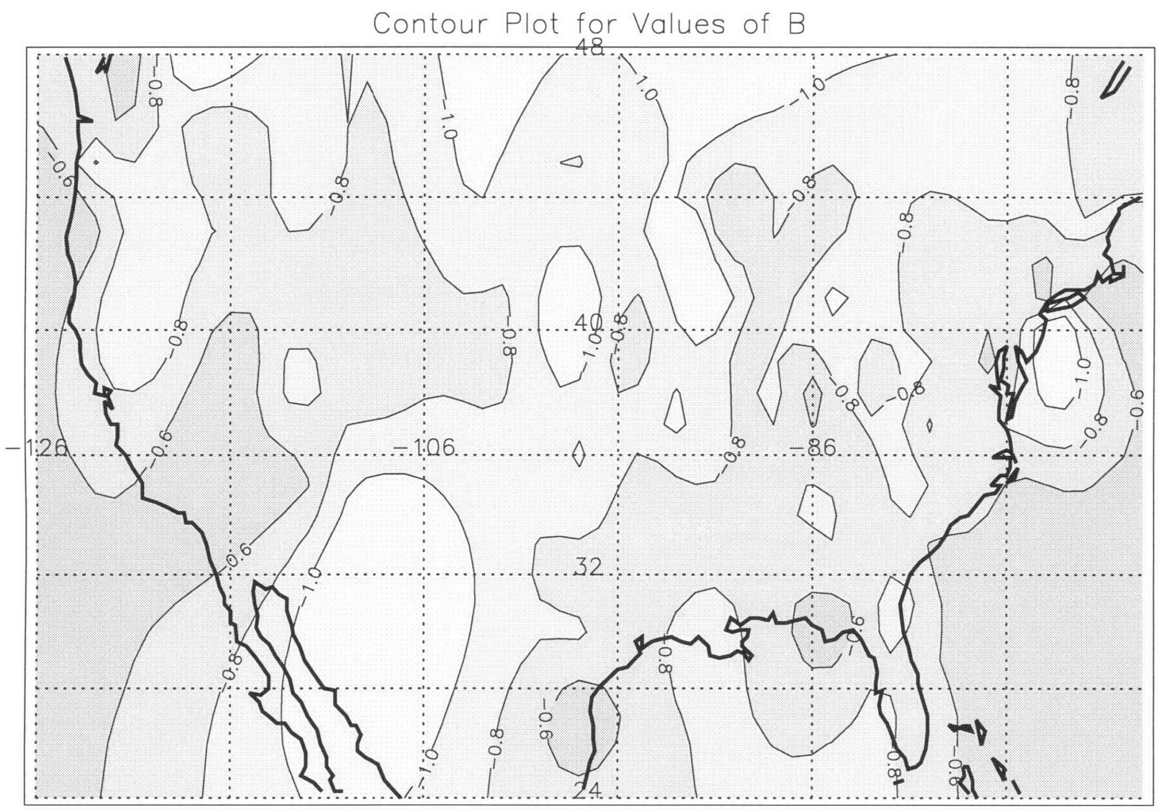

Figure 3-10: Contour plot of the individual station coefficient values of B.

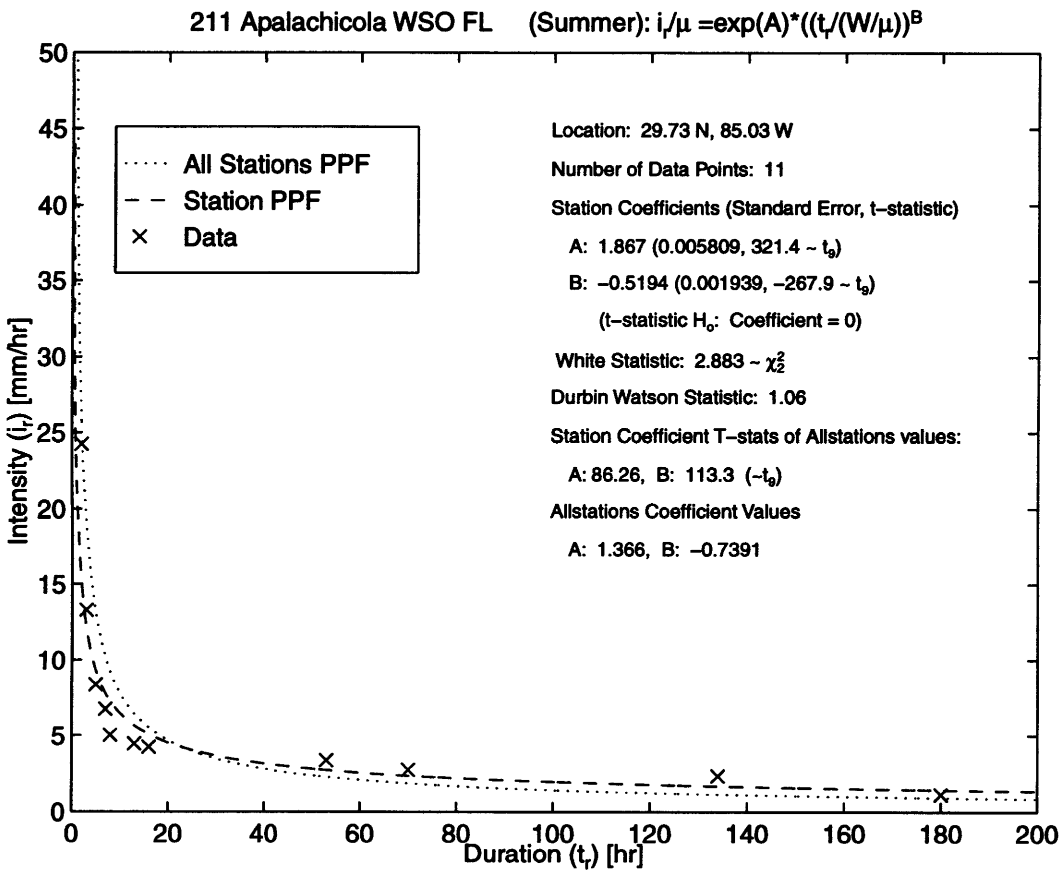

Figure 3-11: General model recovered curve versus the individual station curve at Apalachicola, FL.

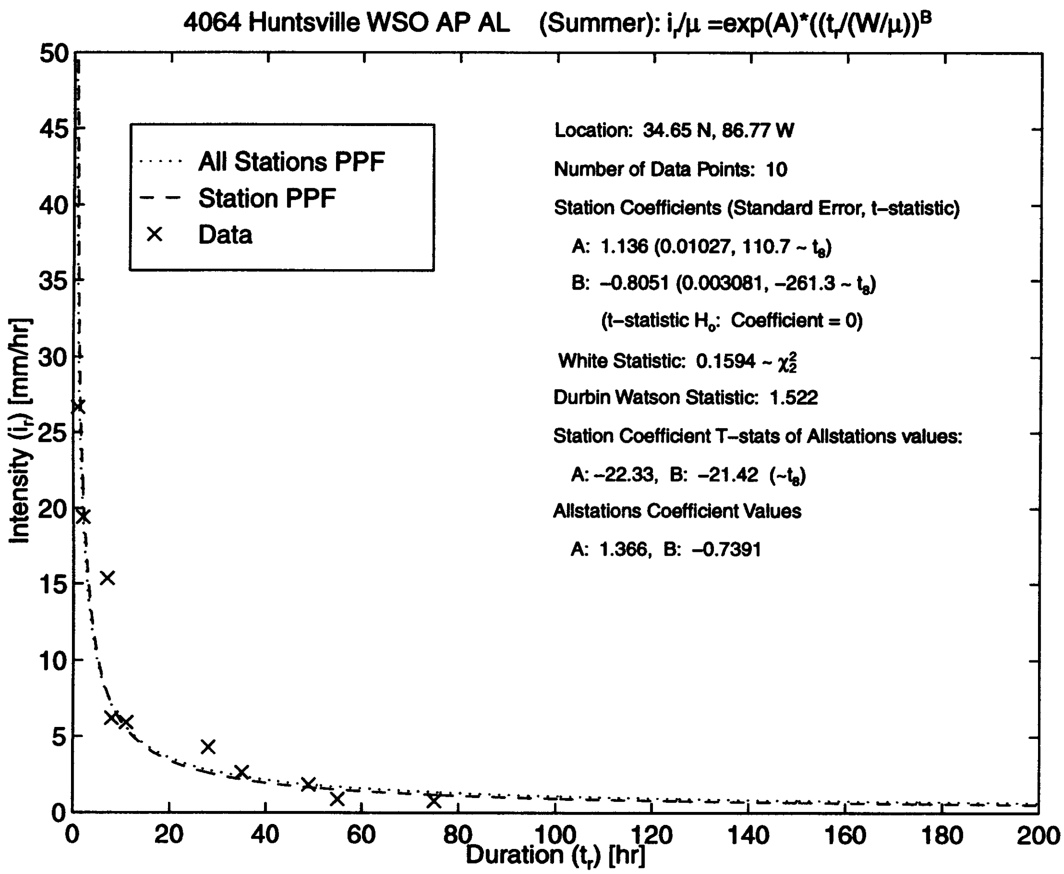

Figure 3-12: General model recovered curve versus the individual station curve at Huntsville, AL.

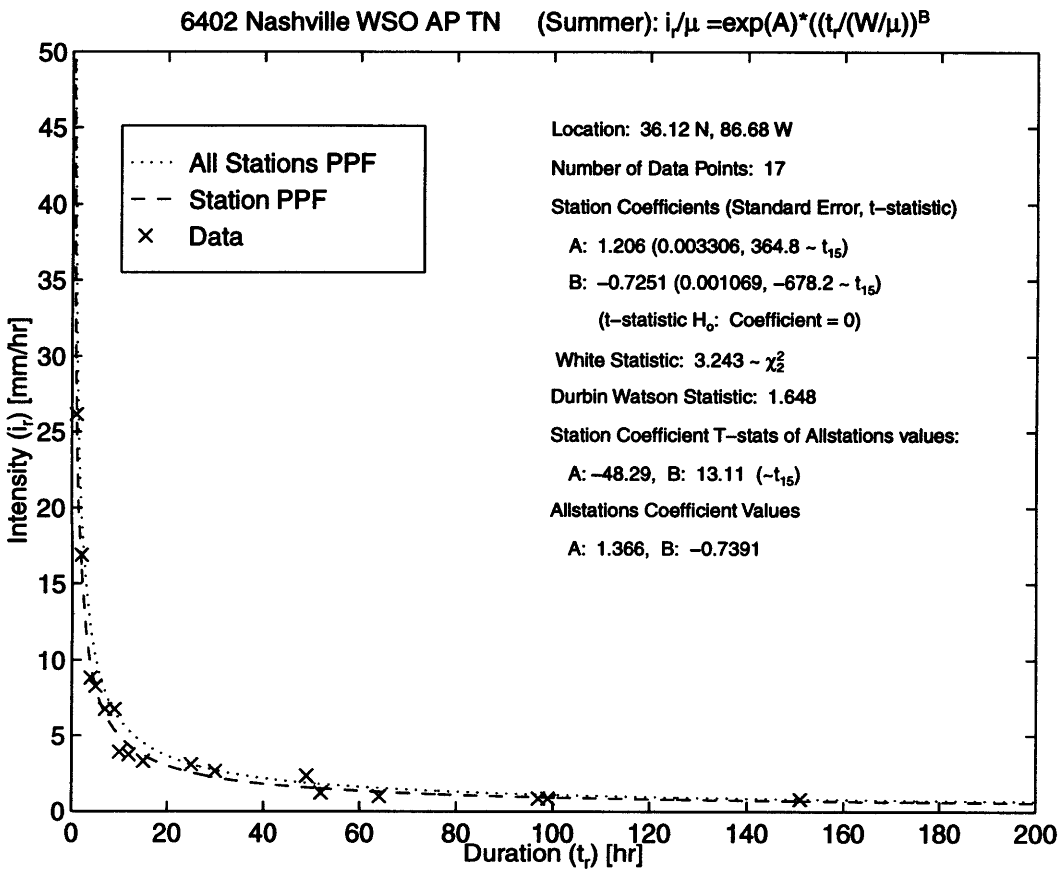

Figure 3-14: General model recovered curve versus the individual station curve at Nasville, TN.

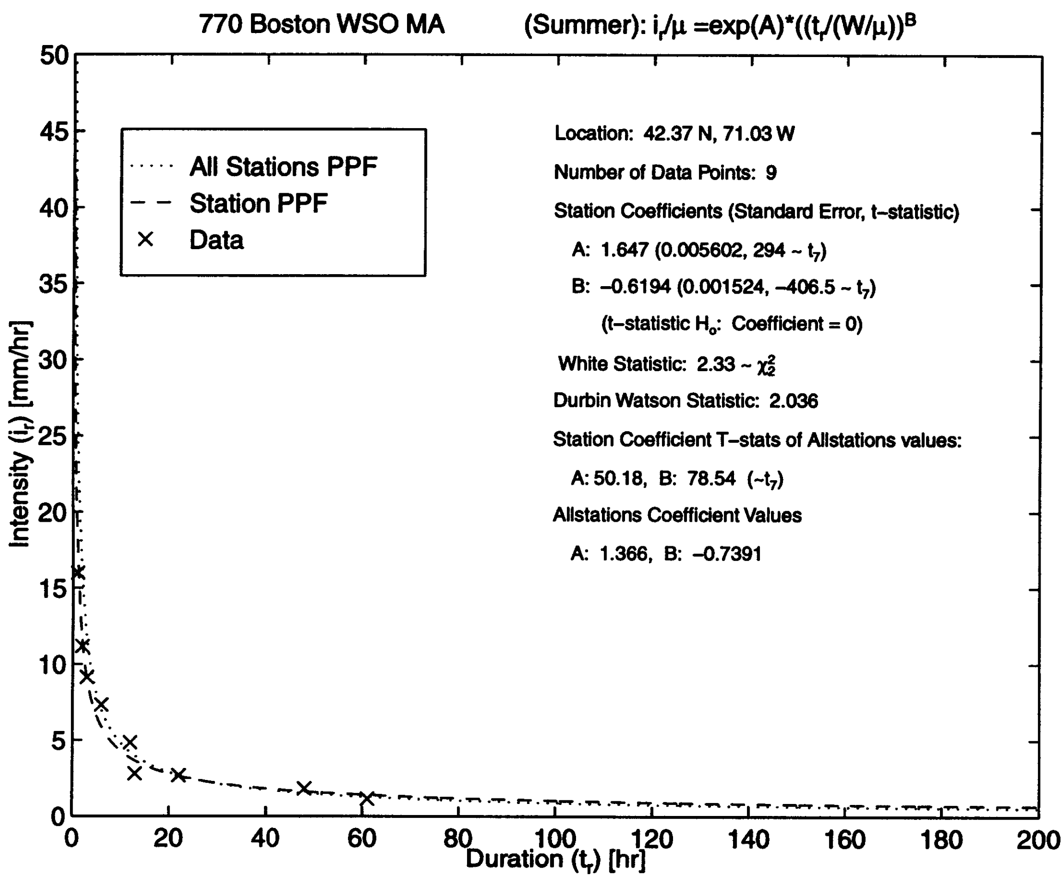

Figure 3-13: General model recovered curve versus the individual station curve at Boston, MA.

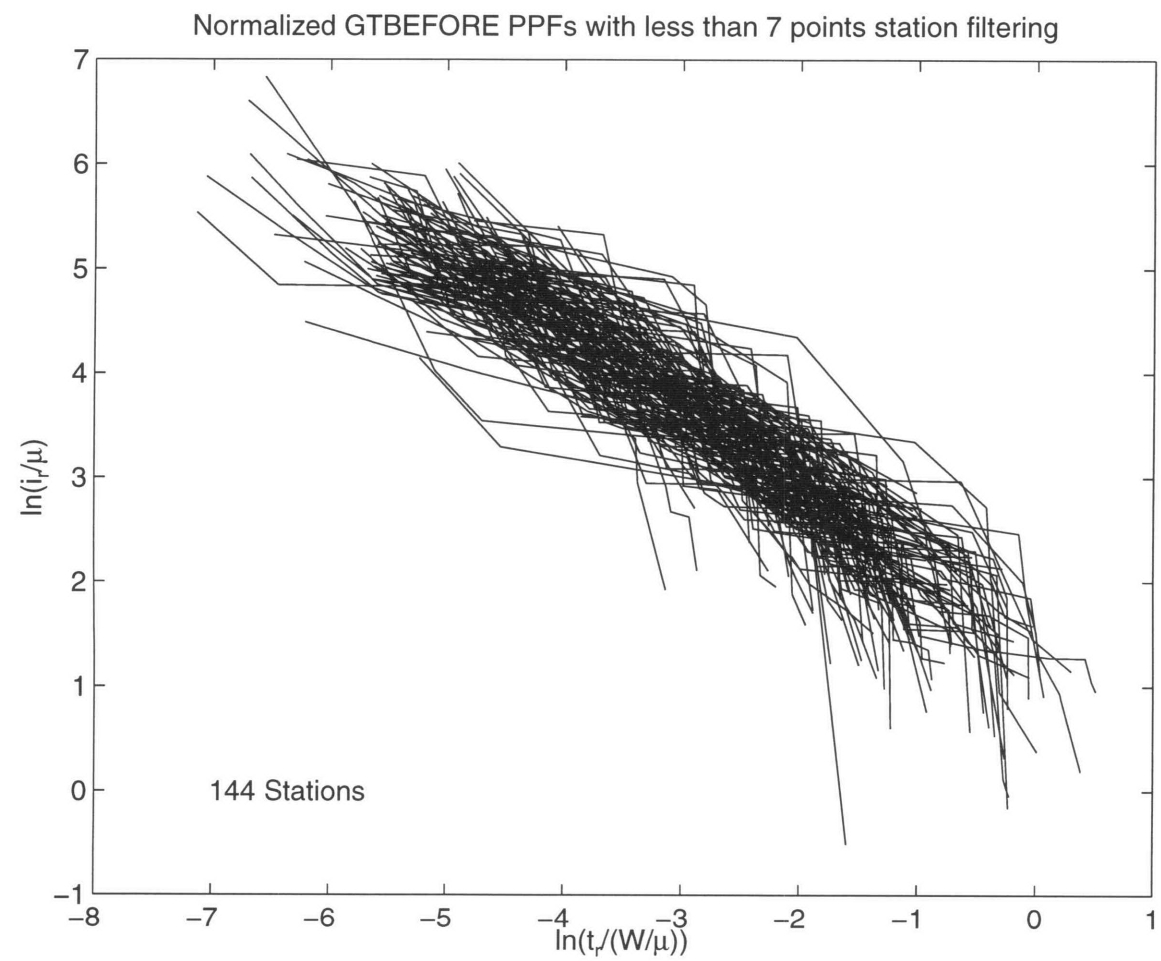

Figure 3-15: Natural log-log plots of the GTBEFORE PPF curves from 144 raingauge stations where i_r and t_r are normalized by stations values of μ and W/μ respectively.

with the regresson coefficients used in Equation 2.11 instead of waiting a sufficient number of years to develop a precipitation event history.

Chapter 4

Future Directions

Because the fitted curve Equation 2.11 is a hyperbolic where the curve goes to infinity at both ends, a problem is posed in quantifying the goodness of fit at the extremes of the curves. From Subsection 3.3.1 in Chapter 3 and from Appendix D, one observes that, in this work, the ways to statistically quantify the goodness of fit of the PPF model are not completely robust. This is primarily due to the differences in the scales and dimensions of the axes, and because the causality direction between the intensity and duration is uncertain. To facilitate the developement of this model, the goodness of fit techniques must be improved upon so that more objective comparisons of various methods used to generate the recovered PPFs can be made.

Another avenue of development for the PPF model lies within the regression process. The assessment, improvement, and development of the equation may lie in the research on theory of storm generation. While general ideas are known about what types of conditions create storms what general type of storms one may expect, the finer detail mechanics of storm production are unknown and/or unsolvable by modern computational methods. Convergence is one such topic of consideration. For intense storms, the convergent flow must be strong to sustain the storm with enough water vapor. Strong convergence, however, is also unstable. Accurate theoretical modelling of convergent flow in a storm may give some insights into storm intensity and durations.

The lack of sufficient storm theory contributes to the possibility that there may be better parameters for I and T than μ and W/μ respectively. A large part of the success of the regression effort is the inclusion of relevant normalizing parameters.

Assuming that consistant, quantitative ways can be found to measure the fit of the the recovered PPF model to the actual data, the next issue to consider is a criteria for a sufficiently good fit. Here the evaluation system must consider three issues:

1. *Performance.* The methods to measure the performance of the PPF model must be able to be reliably and consistantly compared to other possible models.

2. *Scientific.* Ideally, a well performing PPF model will provide insight into the physical relation factors of storms.

3. *Practicality.* The overall goal of the PPF model is to enable viable synthetic precipitation series to be produced on a sufficient accuracy for purposes such as flood hazard control.

Storm Intensity-Duration Variations

A key feature of storm characterization in this work is how the intensity is determined. We assumed earlier that the storm is characterized by a constant i_r value—an average intensity for the entire storm. Obviously, this is an approximation. Intensity fluctuations within a storm can make a difference in hydrologic modeling. An avenue of further research is to develop a model of precipitation pulses within a storm. Earlier in Chapter 2, it was suggested that the various precipitation pulses may not be independent within a storm. Such intuition is based on the fact that, in some respects, a storm is an organized system. This is not to say that the within storm precipitation impulses are completely dependent. There may be a stochastic nature within which is influenced by regional parameters or conditions. Therefore, the intensity and the duration of pulses within a storm may be analyzable in a similar methodology presented by this work. Any precipitation model will be markedly improved if the precipitation within a storm can be accurately simulated.

If the PPF model described in the preceding chapters viable, then the next direction to proceed will be to develop normalized methods of assigning probabilities to the storm combinations in the rainstorm feasibility envelope similar to IFD curves.

The GTBEFORE Algorithm revisited

This work would not be complete without a re-emphasis of the GTBEFORE algorithm. The crux of the PPF model is the GTBEFORE algorithm and the way it selects actual data points for the PPF. GTBEFORE provides consistant methodology and criteria for selecting the proper data points. If such an algorithm were not available, evaluating the goodness of fit of the recovered PPFs would be entirely visual. GTBEFORE provides a database upon which to calibrate and fit the functional forms, and to statistically analyze the fitted functional forms. GTBEFORE may have applications in other field too when boundaries of data envelopes need to be determined. Of course, the assumptions in such applications must be similar to those stated in Appendix E.

Appendix A

Station Locations

The following table lists the one hundred seventy-four stations whose rain gauge measurements were available for processing [5, 33].

Column Descriptions:

- SERID number is an integer number assigned to each individual station for the ease of computational data processing.
- STAT is the raingauge station number.
- Longitude (LONG) and latitude (LAT) are given in degrees and decimals of degrees.

Table A.1: Station names, labels, and locations.

SERID #	STAT #	NAME	STATE	LAT	LONG
1	211	Apalachicola WSO	FL	29.7330	85.0330
2	4570	Key West WSO	FL	24.5500	81.7500
3	5663	Miami WSCMO	FL	25.8000	80.3000
4	8788	Tampa WSCMO	FL	27.9670	82.5330
5	9525	West Palm Bch WSO	FL	26.6830	80.1170
6	2158	Daytona Beach WSO	FL	29.1830	81.0500
7	4358	Jacksonville WSO	FL	30.5000	81.7000
8	8758	Tallahassee WSO	FL	30.3830	84.3670
9	1791	Columbia WSO	MO	38.8170	92.2170
10	7976	Springfield WSO	MO	37.2330	93.3830
11	4064	Huntsville WSO AP	AL	34.6500	86.7670
12	5550	Montgomery WSO AP	AL	32.3000	86.4000
13	3010	Flagstaff WSO AP	AZ	35.1330	111.667
14	6481	Phoenix WSFO AP	AZ	33.4330	112.017
15	8820	Tuscon WSO AP	AZ	32.1330	110.950
16	9439	Winslow WSO	AZ	35.0170	110.733
17	2574	Fort Smith WSO	AR	35.3330	94.3670
18	822	Bishop WSO	CA	37.3670	118.367
19	3257	Fresno WSO	CA	36.7670	119.717
20	5114	Los Angeles WSO	CA	33.9330	118.400
21	7740	San Diego WSO	CA	32.7330	117.167
22	2910	Eureka WSO	CA	40.8000	124.167
23	5983	Mount Shasta	CA	41.3170	122.317
24	7292	Red Bluff WSO	CA	40.1500	122.250
25	897	Blue Canyon WSMO	CA	39.2830	120.700
26	7769	San Francisco WSO	CA	37.6170	122.383
27	7772	San Fran Missi Dolor	CA	37.7670	122.433
28	7846	San Luis Dam	CA	37.0500	121.067
29	2220	Denver WSFO	CO	39.7670	104.8670
30	130	Alamosa WSO	CO	37.4500	105.867
31	3488	Grand Junction WSO	CO	39.1000	108.550
32	806	Bridgeport WSO	CT	41.1670	73.1330
33	3456	Hartford WSO	CT	41.9330	72.6830
34	9595	Wilmington WSO	DE	39.6670	75.6000
35	435	Athens WSO	GA	33.9500	83.3170
36	451	Atlanta WSO	GA	33.6500	84.4330
37	2166	Columbus WSO	GA	32.5170	84.9500
38	5443	Macon WSO	GA	32.7000	83.6500
39	7847	Savannah WSO	GA	32.1330	81.2000
40	7211	Pocatello WSO	ID	42.9170	112.600

SERID #	STAT #	NAME	STATE	LAT	LONG
41	5241	Lewiston WSO	ID	46.3830	117.017
42	1022	Boise WSFO	ID	43.5670	116.217
43	6711	Peoria WSO	IL	40.6670	89.6830
44	7382	Rockford WSO	IL	42.2000	89.1000
45	1166	Cairo WSO	IL	37.0000	89.1670
46	2353	Dixon Springs	IL	37.4330	88.6670
47	8179	Springfield WSO	IL	39.8500	89.6830
48	5751	Moline WSO	IL	41.4500	90.5000
49	3037	Fort Wayne WSO	IN	41.0000	85.2000
50	8187	South Bend WSO	IN	41.7000	86.3170
51	2738	Evansville WB	IN	38.0500	87.5330
52	4259	Indianapolis WSFO	IN	39.7330	86.2670
53	2203	Des Moines WSFO	IA	41.5330	93.6500
54	2367	Dubuque WSO	IA	42.4000	90.7000
55	8315	Traer	IA	42.1830	92.4670
56	8706	Waterloo WSO	IA	42.5500	92.4000
57	7708	Sioux City WSO	IA	42.4000	96.3830
58	1767	Concordia WSO	KS	39.5500	97.6500
59	2164	Dodge City WSO	KS	37.7670	99.9670
60	8167	Topeka WSFO	KS	39.0670	95.6330
61	3153	Goodland WSO	KS	39.3670	101.7000
62	4954	Louisville WSFO	KY	38.1830	85.7330
63	1855	Covington WSO	KY	39.0670	84.6670
64	4746	Lexington WSO	KY	38.0330	84.6000
65	5078	Lake Charles WSO	LA	30.1170	93.2170
66	6660	New Orleans WSCMO	LA	29.9830	90.2500
67	1175	Caribou WSO	ME	46.8670	68.0170
68	6905	Portland WSMO	ME	43.6500	70.3170
69	465	Baltimore WSO	MD	39.1830	76.6670
70	666	Birch Hill WSO	MA	42.6330	72.1170
71	736	Blue Hill WSO	MA	42.2170	71.1170
72	770	Boston WSO	MA	42.3670	71.0330
73	2107	East Brimfield Lake	MA	42.1170	72.1330
74	9923	Worcester WSO	MA	42.2670	71.8670
75	3333	Grand Rapids WSO	MI	42.8830	85.5170
76	2103	Detroit Metro WSO	MI	42.2330	83.3330
77	4641	Lansing WSO	MI	42.7670	84.6000
78	2248	Duluth WSO	MN	46.8330	92.1830
79	5435	Minn-St Paul WSO	MN	44.8830	93.2170
80	7294	St Cloud WSO	MN	45.5500	94.0670
81	7004	Rochester WSO	MN	43.9170	92.5000
82	5776	Meridian WSO	MS	32.3330	88.7500
83	4472	Jackson WSFO	MS	32.3170	90.0830

SERID #	STAT #	NAME	STATE	LAT	LONG
84	807	Billings WSO	MT	45.8000	108.533
85	3751	Great Falls WSCMO	MT	47.4830	111.367
86	4055	Helena WSO	MT	46.6000	112.000
87	5745	Missoula WSO	MT	46.9170	114.083
88	3395	Grand Island WSO	NE	40.9670	98.3170
89	5995	Norfolk WSO	NE	41.9830	97.4330
90	6065	North Platte WSO	NE	41.1330	100.6830
91	7665	Scottsbluff WSO	NE	41.8670	103.6000
92	8760	Valentine WSO	NE	42.8670	100.5500
93	2631	Ely WSO	NV	39.2830	114.850
94	6779	Reno WSFO	NV	39.5000	119.783
95	9171	Winnemucca WSO	NV	40.9000	117.800
96	1683	Concord WSO	NH	43.2000	71.5000
97	5639	Mount Washington	NH	44.2670	71.3000
98	6026	Newark WSO	NJ	40.7000	74.1670
99	1515	Carrizozo	NM	33.6500	105.883
100	5803	NY Inter AP IDLEWIL	NY	40.6500	73.7830
101	5811	New York WB La Guardia	NY	40.7670	73.8670
102	300	Asheville WSO AP	NC	35.4330	82.5500
103	301	Asheville	NC	35.6000	82.5330
104	1690	Charlotte WSO AP	NC	35.2170	80.9330
105	1458	Cape Hatteras WSO	NC	35.2670	75.5500
106	7069	Raleigh Durham WSFO	NC	35.8670	78.7830
107	9457	Wilmington WSO AP	NC	34.2670	77.9000
108	3630	Greensboro WSO AP	NC	36.0830	79.9500
109	2859	Fargo WSO AP	ND	46.9000	96.8000
110	819	Bismarck SWFO AP	ND	46.7670	100.7670
111	2075	Dayton WSCMO AP	OH	39.9000	84.2000
112	1657	Cleveland WSO AP	OH	41.4170	81.8670
113	4865	Mansfield WSO AP	OH	40.8170	82.5170
114	6661	Oklahoma City WSFO A	OK	35.4000	97.6000
115	6546	Pendleton WSO AP	OR	45.6830	118.850
116	328	Astoria WSO AP	OR	46.1500	123.883
117	2709	Eugene WSO AP	OR	44.1170	123.217
118	6751	Portland WSFO AP	OR	45.6000	122.600
119	7500	Salem WSO AP	OR	44.9170	123.017
120	5429	Medford WSO AP	OR	42.3830	122.883
121	7698	Sexton Summit WSO	OR	42.6170	123.367
122	106	Allentown WSO AP	PA	40.6500	75.4330
123	6927	Phoenixville 1 E	PA	40.1170	75.5000
124	9705	W Barre Scrant WSO	PA	41.3330	75.7330
125	6889	Philadelphia WSCMO	PA	39.8830	75.2330
126	6993	Pittsburgh WSCMO2	PA	40.5000	80.2170

SERID #	STAT #	NAME	STATE	LAT	LONG
127	6698	Providence WSO AP	RI	41.7330	71.4330
128	1549	Charleston WSO CI	SC	32.7830	79.9330
129	1544	Charleston WSO AP	SC	32.9000	80.0330
130	3747	Grnvl-Sptnbg WSO AP	SC	34.9000	82.2170
131	4127	Huron WSO AP	SD	44.3830	98.2170
132	7667	Sioux Falls WSFO	SD	43.5670	96.7330
133	6937	Rapid City WSO	SD	44.0500	103.0670
134	6402	Nashville WSO AP	TN	36.1170	86.6830
135	1656	Chattanooga WSO AP	TN	35.0330	85.2000
136	1094	Bristol WSO AP	TN	36.4830	82.4000
137	6750	Oak Ridge ATDL	TN	36.0170	84.2330
138	4950	Knoxville WSO AP	TN	35.8000	84.0000
139	5890	Midland/Odessa WSO	TX	31.9500	102.1830
140	1136	Brownsville WSO AP	TX	25.9000	97.4330
141	2015	Corpus Christi WSO	TX	27.7670	97.5000
142	428	Austin WSO	TX	30.3000	97.7000
143	738	Bertram 3 ENE	TX	30.7500	98.0170
144	4300	Houston WSCMO AP	TX	29.9670	95.3500
145	7945	San Antonio WSFO	TX	29.5330	98.4670
146	9364	Victoria WSO AP	TX	28.8500	96.9170
147	16	Abilene WSO AP	TX	32.4330	99.6830
148	9729	Wichita Falls WSO	TX	33.9670	98.4830
149	7174	Port Arthur WSO	TX	29.9500	94.0170
150	5654	Milford WSMO	UT	38.4330	113.017
151	7598	Salt Lake City	UT	40.7830	111.950
152	5982	North Springfield	VT	43.3330	72.5000
153	8428	Townshend Lake	VT	43.0500	72.7000
154	8556	Union Village Dam	VT	43.8000	72.2670
155	1081	Burlington	VT	44.4670	73.1500
156	5120	Lynchburg WSO	VA	37.3330	79.2000
157	6139	Norfolk WSO	VA	36.9000	76.2000
158	7201	Richmond WSO	VA	37.5000	77.3330
159	7285	Roanoke WSO	VA	37.3170	79.9670
160	8906	Wash Natl WSCMO	VA	38.8500	77.0330
161	7938	Spokane WSO	WA	47.6330	117.533
162	8931	Walla Walla WSO	WA	46.0330	118.333
163	6114	Olympia WSO	WA	46.9670	122.900
164	6858	Quillayute WSCMO	WA	47.9500	124.550
165	7473	Seattle TAC WSCMO	WA	47.4500	122.300
166	8009	Stampede Pass WSCMO	WA	47.2830	121.333
167	9465	Yakima WSO	WA	46.5670	120.533
168	1570	Charleston WSFO	WV	38.3670	81.6000
169	4393	Huntington WSO	WV	38.3670	82.5500

SERID #	STAT #	NAME	STATE	LAT	LONG
170	2718	Elkins WSO	WV	38.8830	79.8500
171	4961	Madison WSO	WI	43.1330	89.3330
172	5479	Milwaukee WSO	WI	42.9500	87.9000
173	3269	Green Bay WSO	WI	44.4830	88.1330
174	8155	Sheridan WSO	WY	44.7670	106.967

Appendix B

Station Data

B.1 Station Statistics and T_{bmin}

The following table describes station statistics for the summer season (June–August). These statistics are, for the most part, used to generate the derived statistics in Section B.2. The various columns are described below:

1. SERID #—Station Series I.D. Number.
2. W—Precipitable water column [mm] [13].[1]
3. μ—Hourly precipitation sample mean [mm/hr].
4. Pr(Rain)—Probability of precipitation for a given time segment of a specified scale [ ]. This statistic is obtained by dividing the total number of time steps in which there was rainfall by the total number of time steps within a specified time period (in this case a season). In this case the Pr(Rain) is for the unit of an hour.
5. σ—Hourly precipitation sample standard deviation [mm/hr].
6. ρ_1—Sample autocorrelation (lag 1) [ ].

[1]See Appendix E for notes on the calculation

7. T_{bmin}—-Breakpoint T_{bmin} [hr].

The Sample autocorrelation function is

$$\rho_k = \frac{\sum_{t=1}^{T-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{T}(x_t - \bar{x})^2}. \tag{B.1}$$

Table B.1: Station statistics.

SERID #	W	μ	Pr(Rain)	σ	ρ_1	T_{bmin}
1	45.226	0.19661	0.05390	1.608	0.3688	24
2	46.827	0.15848	0.04768	1.444	0.2664	25
3	45.796	0.24158	0.06612	1.900	0.3578	37
4	44.354	0.21943	0.05626	1.749	0.4174	22
5	46.450	0.21630	0.05891	1.753	0.2991	44
6	44.563	0.19167	0.05399	1.680	0.3690	23
7	43.056	0.19434	0.05752	1.684	0.1891	24
8	44.246	0.25040	0.05723	1.970	0.3067	23
9	32.805	0.11951	0.03723	1.257	0.3877	19
10	33.950	0.13766	0.03921	1.430	0.3320	12
11	36.629	0.12849	0.03895	1.272	0.2988	13
12	39.698	0.14642	0.03976	1.438	0.4668	25
13	20.153	0.06115	0.03067	0.697	0.5094	48
14	26.451	0.02379	0.00851	0.518	0.1780	28
15	27.346	0.05314	0.02391	0.676	0.2272	23
16	19.400	0.03351	0.01744	0.518	0.2259	33
17	36.003	0.11315	0.03393	1.141	0.3521	26
18	17.696	0.00701	0.00654	0.159	0.9479	42
19	22.163	0.00164	0.00181	0.081	0.6012	6
20	23.520	0.00371	0.00210	0.133	0.9479	80
21	24.312	0.00292	0.00259	0.097	0.9478	12
22	23.614	0.01425	0.01692	0.161	0.6205	6
23	19.023	0.01929	0.01627	0.257	0.6011	14
24	20.401	0.00731	0.00621	0.149	0.5020	11
25	19.674	0.02410	0.01495	0.297	0.7523	24
26	20.513	0.00186	0.00239	0.061	0.6106	4
27	20.646	0.00230	0.00317	0.061	0.6496	10
29	18.014	0.05715	0.03274	0.636	0.3706	22
30	15.708	0.03060	0.02588	0.401	0.2388	26
31	17.160	0.02151	0.01679	0.332	0.0751	29
32	29.894	0.13393	0.04946	1.139	0.3621	8
33	29.340	0.12099	0.05859	0.921	0.2245	7
34	32.002	0.13109	0.05011	1.121	0.1810	9
35	37.407	0.13130	0.04108	1.317	0.2181	26
36	37.812	0.12901	0.04309	1.178	0.2478	23
37	39.182	0.15656	0.04319	1.469	0.4960	23
38	39.847	0.13553	0.04235	1.331	0.4290	22
39	42.069	0.20953	0.05131	1.926	0.1681	24
40	14.559	0.02805	0.02232	0.323	0.6395	25
41	18.119	0.02909	0.03041	0.254	0.4550	20

SERID #	W	μ	Pr(Rain)	σ	ρ_1	T_{bmin}
42	16.287	0.02025	0.01750	0.257	0.4230	22
43	31.350	0.13635	0.04270	1.269	0.4764	16
44	29.494	0.14902	0.04565	1.393	0.3658	5
45	35.360	0.14408	0.03908	1.402	0.2651	32
47	31.760	0.11681	0.04108	1.124	0.4437	16
48	30.150	0.15119	0.04688	1.434	0.6146	12
49	30.091	0.11309	0.04458	1.012	0.2938	11
50	29.601	0.13435	0.04979	1.144	0.4698	8
51	34.032	0.12841	0.04079	1.301	0.3777	13
52	31.441	0.14801	0.04587	1.367	0.4514	11
53	29.715	0.14927	0.04736	1.410	0.5303	15
54	29.154	0.12765	0.04403	1.102	0.5053	12
56	28.582	0.13183	0.04293	1.202	0.3529	6
57	26.944	0.10507	0.03927	1.066	0.5716	5
58	29.078	0.12809	0.03533	1.447	0.2443	8
59	27.670	0.09635	0.02899	1.087	0.3620	21
60	34.286	0.13855	0.03908	1.374	0.5089	12
61	23.005	0.08924	0.02931	1.053	0.1706	56
62	33.074	0.13684	0.04668	1.236	0.4020	11
63	31.766	0.14376	0.05041	1.220	0.3003	11
64	32.849	0.15292	0.04791	1.328	0.4189	9
65	44.033	0.17752	0.04328	1.613	0.4513	23
66	43.840	0.21832	0.05105	1.989	0.2817	23
67	24.073	0.13915	0.08375	0.945	0.5126	5
68	28.182	0.11384	0.06205	0.868	0.6747	6
69	32.771	0.13864	0.04781	1.225	0.5020	9
71	29.201	0.13021	0.06114	1.004	0.4005	8
72	29.109	0.10588	0.05778	0.793	0.5634	8
74	29.128	0.14411	0.06438	1.110	0.6339	7
75	28.405	0.11398	0.04565	1.006	0.3403	9
76	28.486	0.10637	0.04202	1.053	0.4380	10
77	28.123	0.10478	0.04471	1.011	0.4672	7
78	23.755	0.14105	0.06159	1.087	0.5322	5
79	26.752	0.13334	0.05027	1.228	0.6719	10
80	26.535	0.12851	0.04849	1.136	0.4509	29
81	27.253	0.13781	0.05043	1.324	0.4326	5
82	40.817	0.14389	0.03775	1.427	0.5634	24
83	41.883	0.14206	0.03571	1.541	0.4621	23
84	17.030	0.04126	0.03151	0.442	0.3731	24
85	17.085	0.05518	0.04011	0.513	0.6487	14
86	15.484	0.05123	0.03675	0.547	0.7772	20
87	15.438	0.04113	0.03998	0.343	0.4470	26
88	27.281	0.09390	0.03416	1.040	0.2646	25

SERID #	W	μ	Pr(Rain)	σ	ρ_1	T_{bmin}
89	27.172	0.11294	0.04047	1.173	0.5224	13
90	25.653	0.09035	0.03335	1.011	0.3003	27
91	18.948	0.06708	0.03196	0.783	0.6746	25
92	24.704	0.09623	0.03840	1.036	0.4283	27
93	13.292	0.02372	0.02070	0.307	0.4316	30
94	17.114	0.01061	0.01074	0.175	0.8061	27
95	13.932	0.01969	0.01601	0.274	0.4623	28
96	27.131	0.10808	0.05917	0.879	0.5372	7
97	25.072	0.26399	0.13577	1.109	0.8601	16
98	30.343	0.13603	0.05241	1.175	0.4352	8
100	30.597	0.12913	0.04849	1.195	0.5528	10
101	30.338	0.13402	0.05257	1.183	0.3599	10
102	32.670	0.13324	0.05063	1.198	0.4840	26
103	32.404	0.10016	0.04468	0.962	0.2881	30
104	34.995	0.12349	0.04296	1.245	0.3550	26
105	41.819	0.15895	0.05060	1.326	0.3122	11
106	37.316	0.12359	0.04257	1.142	0.4053	16
107	39.876	0.22019	0.05949	1.881	0.3306	25
108	35.593	0.14185	0.04591	1.315	0.5332	25
109	24.333	0.09627	0.03995	1.015	0.1815	5
110	23.739	0.06532	0.03565	0.683	0.0948	8
111	30.867	0.13605	0.05021	1.163	0.6449	9
112	29.459	0.13651	0.05357	1.130	0.4793	9
113	30.112	0.14143	0.04823	1.262	0.4579	8
114	36.288	0.10798	0.02844	1.234	0.6422	25
115	18.569	0.02268	0.02096	0.274	0.7530	19
116	22.233	0.06195	0.06282	0.378	0.7577	22
117	22.385	0.04124	0.03277	0.392	0.5737	13
118	21.899	0.04334	0.04251	0.338	0.5687	19
119	22.211	0.03624	0.03565	0.303	0.8043	18
120	20.630	0.01772	0.01724	0.215	0.5413	23
122	29.355	0.15064	0.05629	1.200	0.3549	6
124	27.540	0.12338	0.06036	0.972	0.4538	8
125	31.863	0.14049	0.04752	1.262	0.3911	7
126	28.890	0.12722	0.05639	0.975	0.3052	10
127	29.495	0.12555	0.05441	1.086	0.5024	7
129	42.252	0.21661	0.05557	1.909	0.2299	28
130	34.372	0.14399	0.04655	1.358	0.2074	26
131	27.275	0.08600	0.03555	0.892	0.4755	49
132	25.919	0.10979	0.04202	1.061	0.3501	8
133	21.975	0.08105	0.03827	0.894	0.3711	29
134	36.631	0.13690	0.04095	1.297	0.5026	24
135	35.103	0.13514	0.04629	1.304	0.4165	23

SERID #	W	μ	Pr(Rain)	σ	ρ_1	T_{bmin}
136	31.738	0.13125	0.04972	1.089	0.3973	18
137	33.032	0.14466	0.04853	1.244	0.3299	20
138	32.960	0.13502	0.04364	1.235	0.3894	21
139	29.441	0.05776	0.02077	0.840	0.3349	40
140	43.417	0.10261	0.03115	1.147	0.4429	25
141	42.597	0.11990	0.03025	1.378	0.3460	26
142	38.146	0.09071	0.02750	1.137	0.6735	25
144	40.849	0.14991	0.03863	1.596	0.3491	24
145	38.747	0.10385	0.03157	1.146	0.4058	26
146	41.306	0.13485	0.03801	1.418	0.3733	31
147	32.758	0.07781	0.02491	0.976	0.4910	30
148	35.206	0.08474	0.02417	1.025	0.3836	25
149	42.801	0.18708	0.04539	1.734	0.1153	29
150	14.967	0.02772	0.01782	0.385	0.2963	29
151	16.222	0.02748	0.01895	0.340	0.5545	23
155	24.808	0.12874	0.06981	0.944	0.6008	8
156	33.236	0.14270	0.04995	1.306	0.5289	27
157	37.232	0.14596	0.04613	1.348	0.2131	8
158	35.239	0.14074	0.04578	1.327	0.4519	21
159	32.307	0.13046	0.04849	1.176	0.5342	23
160	33.721	0.12960	0.04597	1.185	0.4375	8
161	17.404	0.03285	0.03073	0.326	0.8068	9
162	18.745	0.02858	0.02578	0.263	0.5080	18
163	21.761	0.04978	0.04791	0.381	0.7346	23
164	20.624	0.10978	0.09715	0.559	0.7871	20
165	21.401	0.04276	0.04186	0.307	0.7579	34
166	20.327	0.10851	0.11154	0.484	0.8679	21
167	19.668	0.01479	0.01398	0.245	0.7578	7
168	32.447	0.15792	0.05875	1.290	0.4944	13
169	33.422	0.14174	0.05477	1.162	0.4633	13
170	30.840	0.16569	0.07486	1.072	0.6029	11
171	28.405	0.13198	0.04697	1.227	0.3344	6
172	28.593	0.11524	0.04814	0.974	0.5080	7
173	27.341	0.11730	0.04982	0.996	0.3750	26
174	17.101	0.04259	0.03138	0.445	0.5690	27

B.2 Derived Station Statistics

The previous table is used to derive the following statistics. The statistics below are conceived to normalize parameters I and T for i_r and t_r. Therefore all the statistics have units of either [hr] or [mm/hr]. The columns are:

1. SERID #—Station Series I.D. Number

2. $\frac{W}{\mu}$—Residence time [hr]. This is determined by dividing the precipitable water by the precipitation sample mean [hr]. This variable represents the average circulation time of water in an air column.

3. $\frac{\mu}{Pr(Rain)}$ —Conditional rainstorm intensity: the sample precipitation mean divided by the probability of rain [mm/hr]. This is a representation of the sample mean precipitation rate when it actually is raining. A low probability of rain implies that the rainfall is concentrated into a smaller total time. For the same precipitation sample mean, the mean precipitation when raining is higher if the probability of rain is low, than if the probability of rain was very high (i.e. it rains all the time).

4. E(i_r) – Rectangular Pulses Model expected value of i_r [mm/hr].

5. E(t_r) – Rectangular Pulses Model expected value of t_r [hr].

6. E(t_b) – Rectangular Pulses Model expected value of $t_b[hr]$.

The Poisson expected value statistics are derived from the Rectangular Pulses Model (RPM) [20, 25, 26]. The formulation of the derived statistics from the model is not expanded here. Instead the reader is referred to works by Ignacio Rodriguez-Iturbe and Steven Margulis [20, 25] for more detailed discussions.

The Rectangular Pulses Model does make some distributional assumptions about i_r and t_r [20, 25, 26], and if, indeed, rainfall is a Poisson process, then the expected value of the RPM storm parameters can be found given μ, ρ_1, and σ^2. E(t_r) is determined from ρ_1 using a bisection numerical solution method. The general relationship between E(t_r) and ρ_1 is

$$\rho_{1,T} = \frac{\left(1 - \exp\left(-\frac{T}{E(t_r)}\right)\right)^2}{2\left(\frac{T}{E(t_r)} - 1 + \exp\left(-\frac{T}{E(t_r)}\right)\right)} \tag{B.2}$$

where T is the level of aggregation in hours. In the case of the analysis of the thesis, $T = 1$ hour. Two other relations hold:

1. $$\mu_T = T\frac{E(i_r)E(t_r)}{E(t_b)} \tag{B.3}$$

   where μ_T is the expected value of rainfall aggregated over period T of the RPM.

2. The variance of rainfall aggregated over period T is,

$$\sigma_T{}^2 = 4\frac{E^3(t_r)E^2(i_r)}{E(t_b)}\left(\frac{T}{E(t_r)} - 1 + \exp\left(-\frac{T}{E(t_r)}\right)\right) \tag{B.4}$$

Placing equation B.3 into equation B.4, E(t_b) is obtained by the equation

$$E(t_b) = \frac{\sigma_T{}^2 T^2}{4E(t_r)\left(\frac{T}{E(t_r)} - 1 + \exp\left(-\frac{T}{E(t_r)}\right)\right)E^2(x_t)} \tag{B.5}$$

and as a result from equation B.3,

$$E(i_r) = \frac{\mu_T E(t_b)}{TE(t_r)}. \tag{B.6}$$

One interesting observation is the dependence of some of the statistics upon the the time unit step T. The statistics change depending upon the the time unit size (or level of aggregation) [20, 25, 29]. As a simple example, assume that it rains three hours out of a six hour period. The probability of rain on an hourly basis is 60 percent while it is 100 percent on a six hour basis.

Table B.2: Derived Station Statistics.

SERID #	$\frac{W}{\mu}$	$\frac{\mu}{Pr(Rain)}$	E(i_r)	E(t_r)	E(t_b)
1	230.025	3.64808	10.904	0.570	31.616
2	295.479	3.32353	12.997	0.400	32.814
3	189.574	3.65342	12.591	0.550	28.660
4	202.134	3.90058	10.842	0.667	32.952
5	214.748	3.67176	13.170	0.451	27.440
6	232.493	3.55004	12.207	0.571	36.337
7	221.553	3.37872	17.476	0.290	26.098
8	176.701	4.37556	14.184	0.463	26.210
9	274.488	3.20972	10.684	0.606	54.185
10	246.630	3.51094	13.027	0.505	47.757
11	285.082	3.29885	11.690	0.450	40.956
12	271.116	3.68291	10.368	0.781	55.272
13	329.572	1.99394	5.575	0.895	81.625
14	1111.960	2.79594	14.029	0.275	162.206
15	514.617	2.22273	9.281	0.343	59.935
16	578.972	1.92169	8.652	0.341	88.121
17	318.199	3.33420	9.776	0.540	46.625
18	2524.772	1.07253	1.859	12.378	3282.296
19	13553.964	0.90241	2.574	1.220	1919.573
20	6332.744	1.76606	2.452	12.378	8171.133
21	8334.481	1.12712	1.666	12.360	7057.846
22	1657.341	0.84213	1.151	1.307	105.583
23	986.409	1.18516	2.219	1.219	140.321
24	2789.740	1.17743	2.136	0.874	255.366
25	816.353	1.61248	2.110	2.255	197.415
26	11046.144	0.77569	1.302	1.261	883.993
27	8973.543	0.72578	0.994	1.456	629.203
29	315.205	1.74564	5.856	0.574	58.774
30	513.330	1.18237	5.514	0.360	64.810
31	797.697	1.28123	11.266	0.131	68.438
32	223.210	2.70762	8.113	0.558	33.783
33	242.506	2.06508	7.619	0.339	21.373
34	244.122	2.61608	11.783	0.279	25.086
35	284.904	3.19582	14.572	0.330	36.668
36	293.088	2.99404	11.044	0.373	31.911
37	250.261	3.62526	9.812	0.857	53.731
38	294.011	3.20051	10.030	0.692	51.217
39	200.778	4.08385	22.741	0.262	28.386
40	518.973	1.25678	2.330	1.402	116.443
41	622.893	0.95655	1.652	0.752	42.714

SERID #	$\frac{W}{\mu}$	$\frac{\mu}{Pr(Rain)}$	E(i_r)	E(t_r)	E(t_b)
42	804.133	1.15734	2.526	0.679	84.683
43	229.919	3.19308	8.581	0.805	50.667
44	197.916	3.26471	10.843	0.564	41.072
45	245.426	3.68676	13.499	0.398	37.313
47	271.895	2.84322	8.146	0.725	50.570
48	199.418	3.22539	8.684	1.279	73.481
49	266.082	2.53688	8.484	0.442	33.175
50	220.337	2.69842	7.124	0.788	41.795
51	265.038	3.14772	10.799	0.587	49.356
52	212.422	3.22659	9.437	0.743	47.383
53	199.070	3.15177	9.161	0.959	58.834
54	228.389	2.89927	6.710	0.884	46.454
56	216.805	3.07100	9.303	0.541	38.187
57	256.440	2.67535	7.155	1.101	74.967
58	227.018	3.62580	16.933	0.368	48.599
59	287.176	3.32406	10.275	0.558	59.465
60	247.473	3.54525	9.565	0.894	61.721
61	257.779	3.04492	15.829	0.265	46.995
62	241.708	2.93130	8.848	0.635	41.043
63	220.963	2.85176	9.588	0.452	30.174
64	214.805	3.19190	8.948	0.670	39.208
65	248.038	4.10140	10.951	0.743	45.828
66	200.806	4.27680	17.364	0.423	33.667
67	172.999	1.66145	4.490	0.905	29.190
68	247.568	1.83469	4.031	1.606	56.861
69	236.367	2.89969	7.661	0.874	48.298
71	224.272	2.12959	6.148	0.632	29.828
72	274.918	1.83262	3.958	1.071	40.021
74	202.131	2.23852	5.376	1.373	51.207
75	249.219	2.49695	7.687	0.519	34.997
76	267.804	2.53119	7.912	0.712	52.980
77	268.396	2.34370	7.156	0.782	53.383
78	168.416	2.29001	5.750	0.965	39.320
79	200.622	2.65243	6.905	1.588	82.207
80	206.482	2.65017	7.503	0.742	43.327
81	197.756	2.73254	9.705	0.700	49.290
82	283.662	3.81156	9.434	1.071	70.196
83	294.825	3.97774	12.332	0.769	66.770
84	412.786	1.30938	3.902	0.578	54.676
85	309.596	1.37570	2.960	1.451	77.839
86	302.273	1.39377	3.314	2.558	165.487
87	375.374	1.02855	2.146	0.733	38.246
88	290.530	2.74874	11.415	0.397	48.311

SERID #	$\frac{W}{\mu}$	$\frac{\mu}{Pr(Rain)}$	E(i_r)	E(t_r)	E(t_b)
89	240.589	2.79073	8.446	0.934	69.865
90	283.917	2.70899	10.467	0.452	52.407
91	282.464	2.09883	5.569	1.605	133.254
92	256.722	2.50601	8.563	0.691	61.457
93	560.330	1.14577	3.030	0.698	89.118
94	1613.282	0.98771	1.617	3.006	458.225
95	707.654	1.22948	2.823	0.770	110.337
96	251.015	1.82675	4.885	0.981	44.319
97	94.973	1.94436	2.510	4.337	41.238
98	223.057	2.59565	7.732	0.706	40.119
100	236.952	2.66287	7.448	1.033	59.581
101	226.373	2.54935	8.768	0.554	36.234
102	245.203	2.63169	7.770	0.825	48.099
103	323.514	2.24206	8.733	0.433	37.780
104	283.383	2.87449	10.632	0.545	46.919
105	263.103	3.14156	10.023	0.472	29.747
106	301.935	2.90308	8.331	0.641	43.240
107	181.094	3.70132	14.126	0.502	32.218
108	250.926	3.09003	8.366	0.968	57.081
109	252.757	2.40961	13.133	0.280	38.160
110	363.402	1.83238	13.269	0.160	32.477
111	226.886	2.70972	6.192	1.431	65.106
112	215.808	2.54815	6.777	0.813	40.339
113	212.912	2.93214	8.351	0.759	44.812
114	336.067	3.79723	8.809	1.416	115.511
115	818.794	1.08184	1.907	2.262	190.224
116	358.891	0.98607	1.328	2.315	49.632
117	542.782	1.25849	2.466	1.108	66.291
118	505.227	1.01967	1.749	1.090	43.986
119	612.941	1.01645	1.410	2.975	115.810
120	1163.972	1.02794	1.769	0.994	99.232
122	194.868	2.67619	8.086	0.545	29.239
124	223.205	2.04398	5.699	0.749	34.599
125	226.795	2.95636	9.116	0.613	39.773
126	227.081	2.25627	6.858	0.460	24.811
127	234.916	2.30743	6.640	0.875	46.287
129	195.056	3.89823	18.034	0.347	28.885
130	238.705	3.09314	14.557	0.315	31.883
131	317.156	2.41887	6.726	0.803	62.782
132	236.070	2.61271	8.737	0.536	42.667
133	271.125	2.11788	8.155	0.574	57.803
134	267.575	3.34272	8.682	0.876	55.542
135	259.748	2.91930	9.798	0.665	48.213

SERID #	$\frac{W}{\mu}$	$\frac{\mu}{Pr(Rain)}$	$E(i_r)$	$E(t_r)$	$E(t_b)$
136	241.815	2.63964	7.210	0.625	34.350
137	228.347	2.98110	9.408	0.501	32.585
138	244.116	3.09395	9.114	0.610	41.150
139	509.750	2.78090	10.666	0.510	94.115
140	423.120	3.29381	9.670	0.723	68.169
141	355.267	3.96406	13.579	0.529	59.892
142	420.502	3.29907	8.690	1.598	153.090
144	272.494	3.88101	14.514	0.534	51.729
145	373.121	3.28904	9.977	0.643	61.744
146	306.320	3.54758	12.293	0.579	52.745
147	420.975	3.12393	8.750	0.844	94.870
148	415.459	3.50675	10.083	0.598	71.176
149	228.784	4.12193	26.173	0.189	26.457
150	539.857	1.55533	4.978	0.446	80.119
151	590.399	1.44977	2.835	1.039	107.202
155	192.699	1.84415	4.466	1.218	42.264
156	232.901	2.85701	8.228	0.954	55.029
157	255.089	3.16395	13.916	0.323	30.833
158	250.384	3.07456	9.341	0.744	49.410
159	247.638	2.69034	7.259	0.971	54.023
160	260.181	2.81940	8.227	0.711	45.132
161	529.795	1.06896	1.799	3.019	165.314
162	655.922	1.10842	1.704	0.891	53.131
163	437.159	1.03899	1.703	2.074	70.979
164	187.868	1.13001	1.606	2.698	39.456
165	500.474	1.02150	1.263	2.317	68.423
166	187.325	0.97284	1.157	4.619	49.268
167	1329.811	1.05834	2.340	2.317	366.544
168	205.467	2.68814	7.514	0.853	40.580
169	235.794	2.58803	7.022	0.772	38.251
170	186.124	2.21343	4.466	1.227	33.077
171	215.224	2.80974	9.963	0.509	38.409
172	248.115	2.39404	5.788	0.891	44.757
173	233.077	2.35460	6.953	0.582	34.486
174	401.531	1.35722	3.089	1.091	79.138

Appendix C

Regression Results and Figures

C.1 Overview

Section C.2 shows the plots of the regression processes described in Chapter 2 and describes the various numerical values listed on the plots. Section C.3 gives an extensive discussion of the regression test statistics

C.2 Regression Plots

Recall, from Chapter 2, that the basic equation for the PPF is

$$\frac{i_r}{I} = C\left(\frac{t_r}{T}\right)^B \tag{C.1}$$

where C is a constant, and I and T are normalization constants of dimensions [mm/hr] and [hr] respectively. Because Equation C.1 is natural log-linear, the regression is run upon the equation

$$\ln\left(\frac{i_{r(jn)}}{I_n}\right) = A + B\ln\left(\frac{t_{r(jn)}}{T_n}\right) + \varepsilon_{jn} \tag{C.2}$$

where n is the n^{th} GTBEFORE point of station j, exp(A) equals C (therefore C is positive), and ε is assumed to be normally distributed $N \sim (\mu = 0, \sigma = 1)$. An ordinary least squares regression is run upon Equation C.2. I and T represent the various statistics computed in Appendix B. On each figure the following statistics for the OLS regression are given.

- Estimated coefficients with the individual standard errors and t-statistics which test the null hypothesis $H_\circ$ that the individual coefficient is equal to zero.
- R^2 for the regression of the natural log-linear Equation C.2.
- Regression F-statistic which tests the null hypothesis that all the coefficients except the regression constant are simultaneously equal to zero.
- P-value of the regression F-statistic.
- White Statistic $\sim (\chi_2^2)$.

For the purpose of visual analysis, the number of natural log cycles on each axis has been kept constant from figure to figure. As a result, the data has been compressed into to equal visual scaling across the models, enabling direct visual comparison of how well the models fit.

Table C.1: Summary of regression coefficients A and B for the various I and T parameters.

I [mm/hr]	T[hr]	A	B
μ	(W/μ)	1.3657	-0.7391
μ	$E(t_r)$	5.5757	-0.6618
μ	$E(t_b)$	2.6808	-0.7086
μ	$T_{bmin(BP)}$	3.5108	-0.7129
$(\mu/Pr(Rain))$	(W/μ)	-1.5143	-0.6161
$(\mu/Pr(Rain))$	$E(t_r)$	2.2638	-0.6377
$(\mu/Pr(Rain))$	$E(t_b)$	-0.4140	-0.5872
$(\mu/Pr(Rain))$	$T_{bmin(BP)}$	0.2739	-0.6736
$E(i_r)$	(W/μ)	-2.6421	-0.6412
$E(i_r)$	$E(t_r)$	1.4553	-0.7167
$E(i_r)$	$E(t_b)$	-1.5129	-0.6246
$E(i_r)$	$T_{bmin(BP)}$	-0.7812	-0.6602
σ	(W/μ)	-0.7449	-0.6852
σ	$E(t_r)$	3.3058	-0.6608
σ	$E(t_b)$	0.4798	-0.6522
σ	$T_{bmin(BP)}$	1.2439	-0.6913

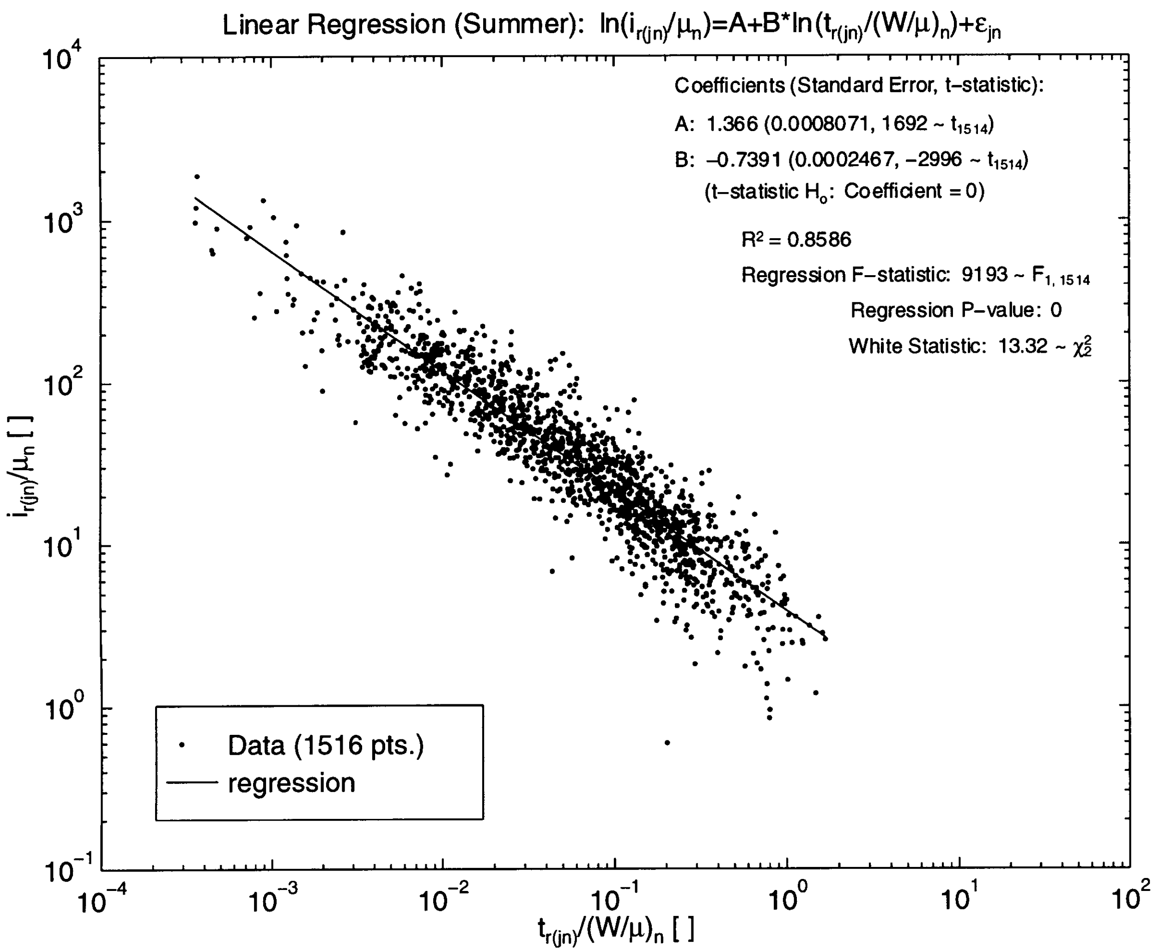

Figure C-1: Ordinary Least Squares Regression on Equation C.2 with $I = \mu$ and $T = W/\mu$.

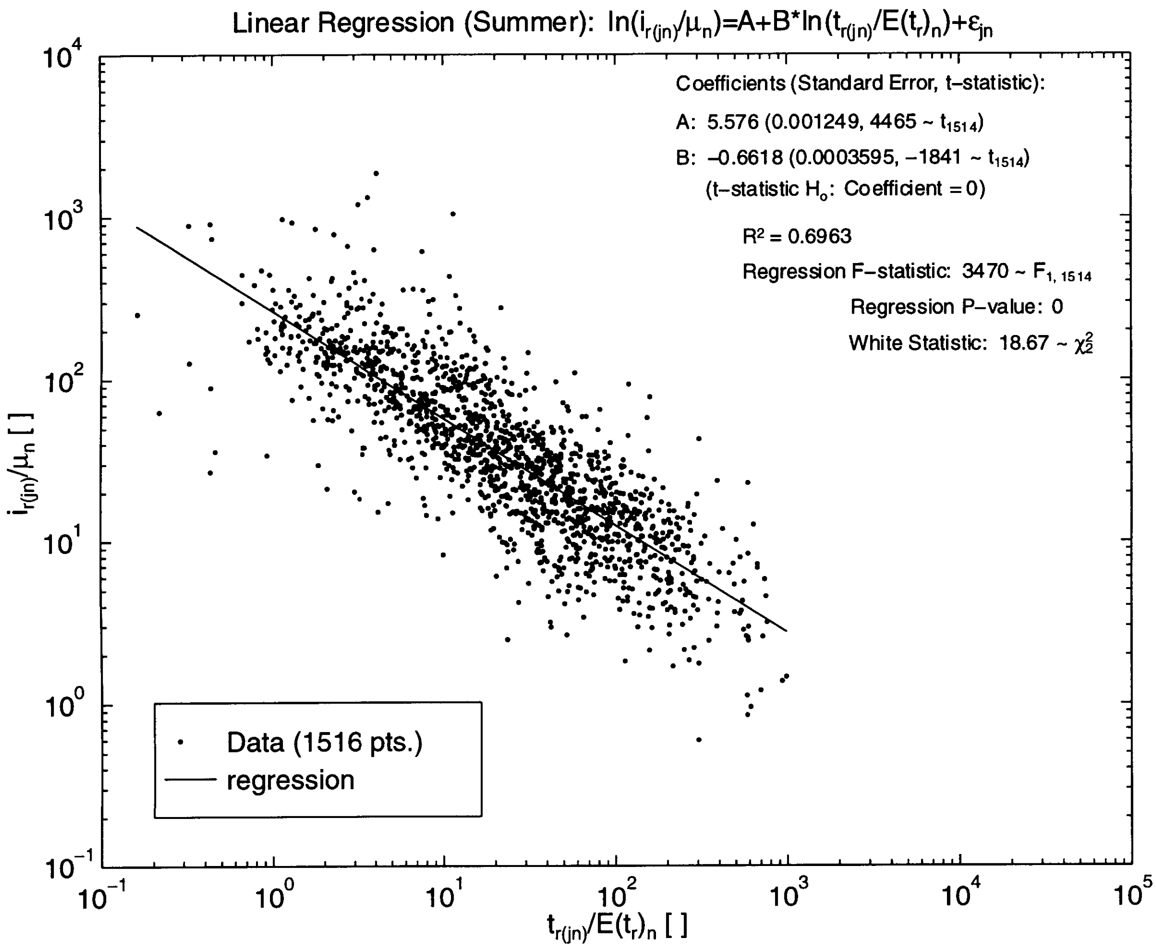

Figure C-2: Ordinary Least Squares Regression on Equation C.2 with $I = \mu$ and $T = E(t_r)$.

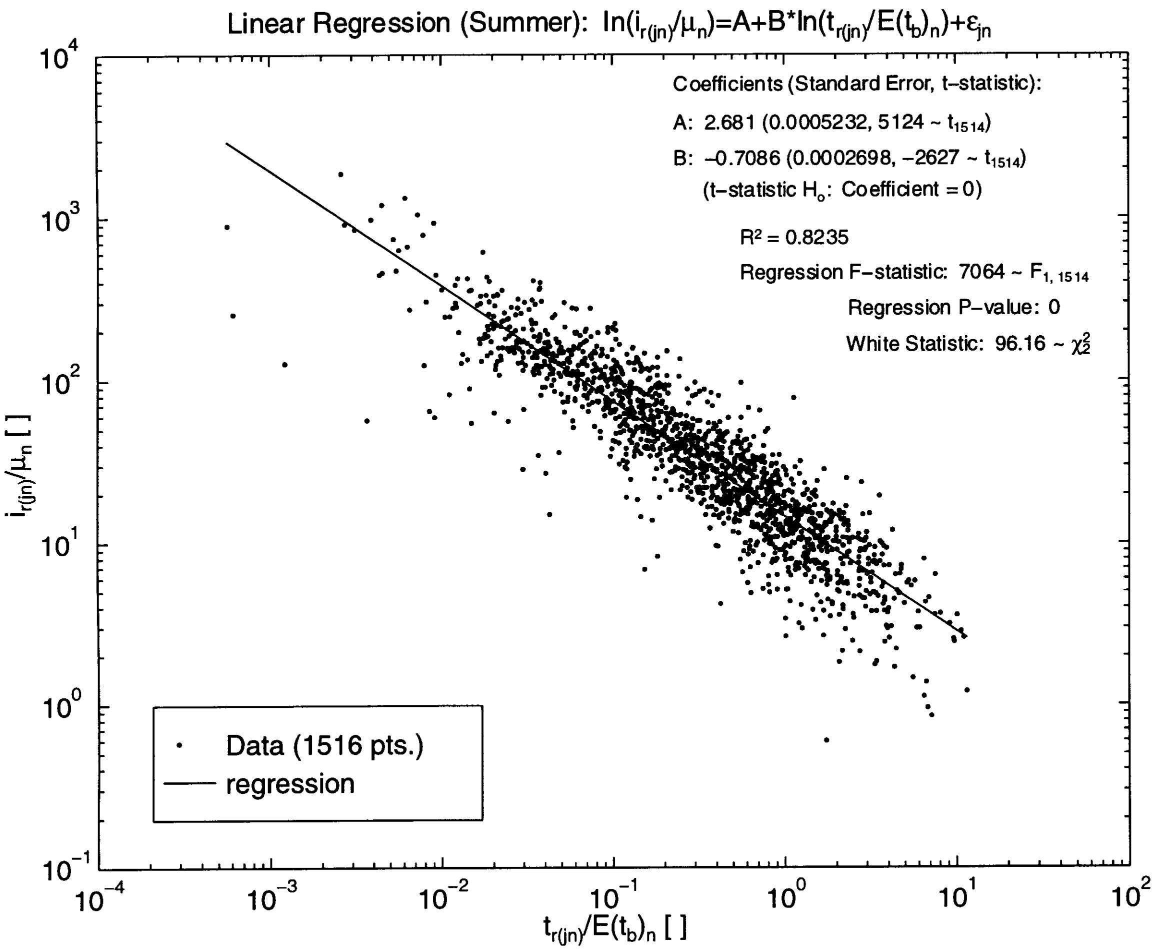

Figure C-3: Ordinary Least Squares Regression on Equation C.2 with $I = \mu$ and $T = E(t_b)$.

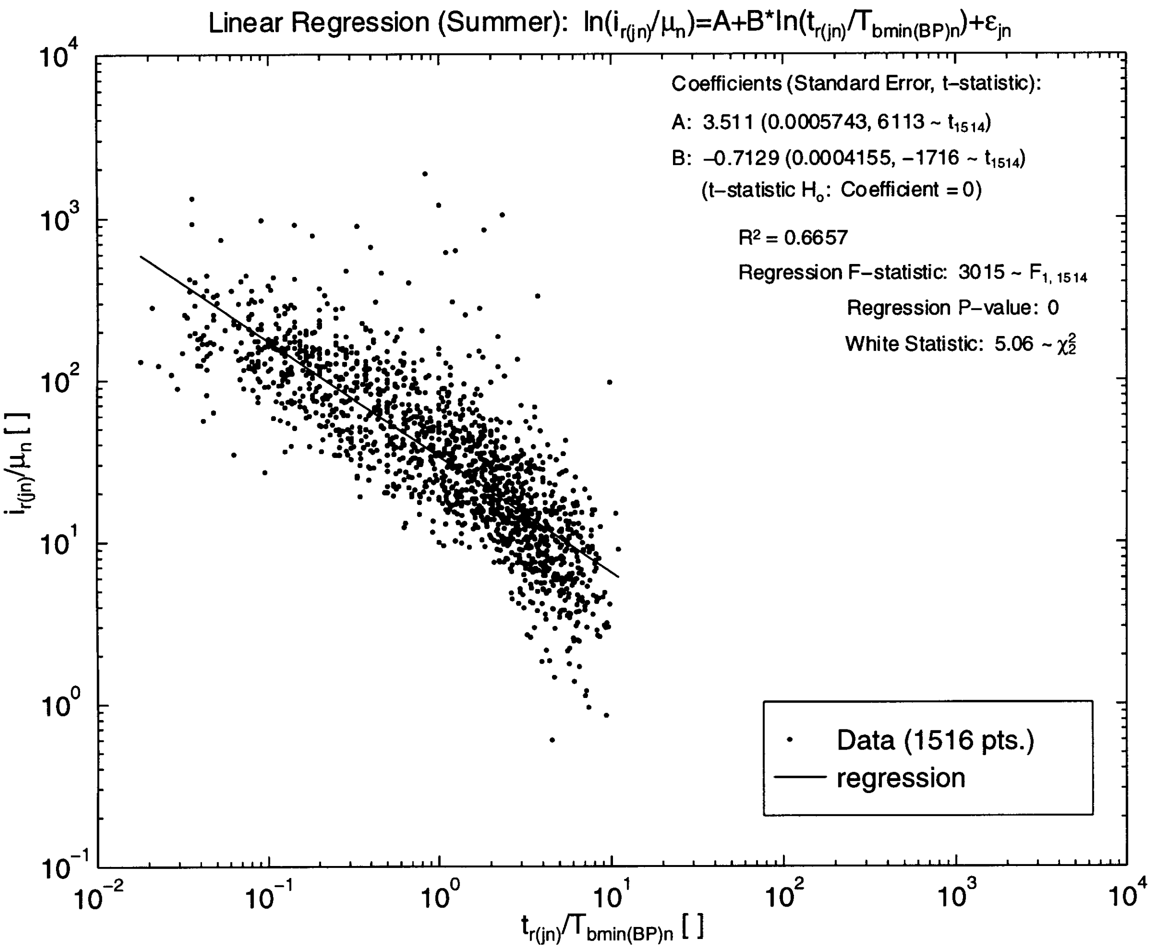

Figure C-4: Ordinary Least Squares Regression on Equation C.2 with $I = \mu$ and $T = T_{bmin(BP)}$.

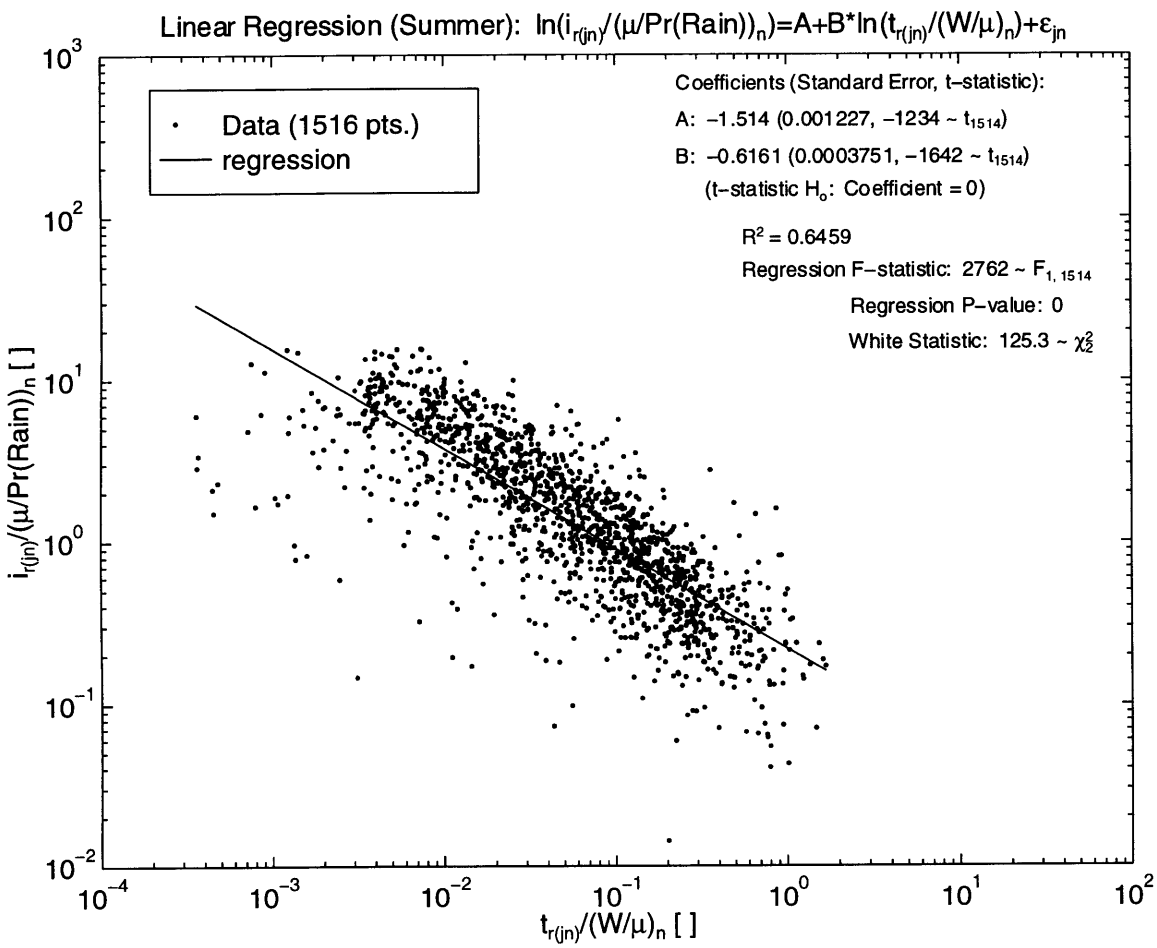

Figure C-5: Ordinary Least Squares Regression on Equation C.2 with $I = \mu/Pr(Rain)$ and $T = W/\mu$.

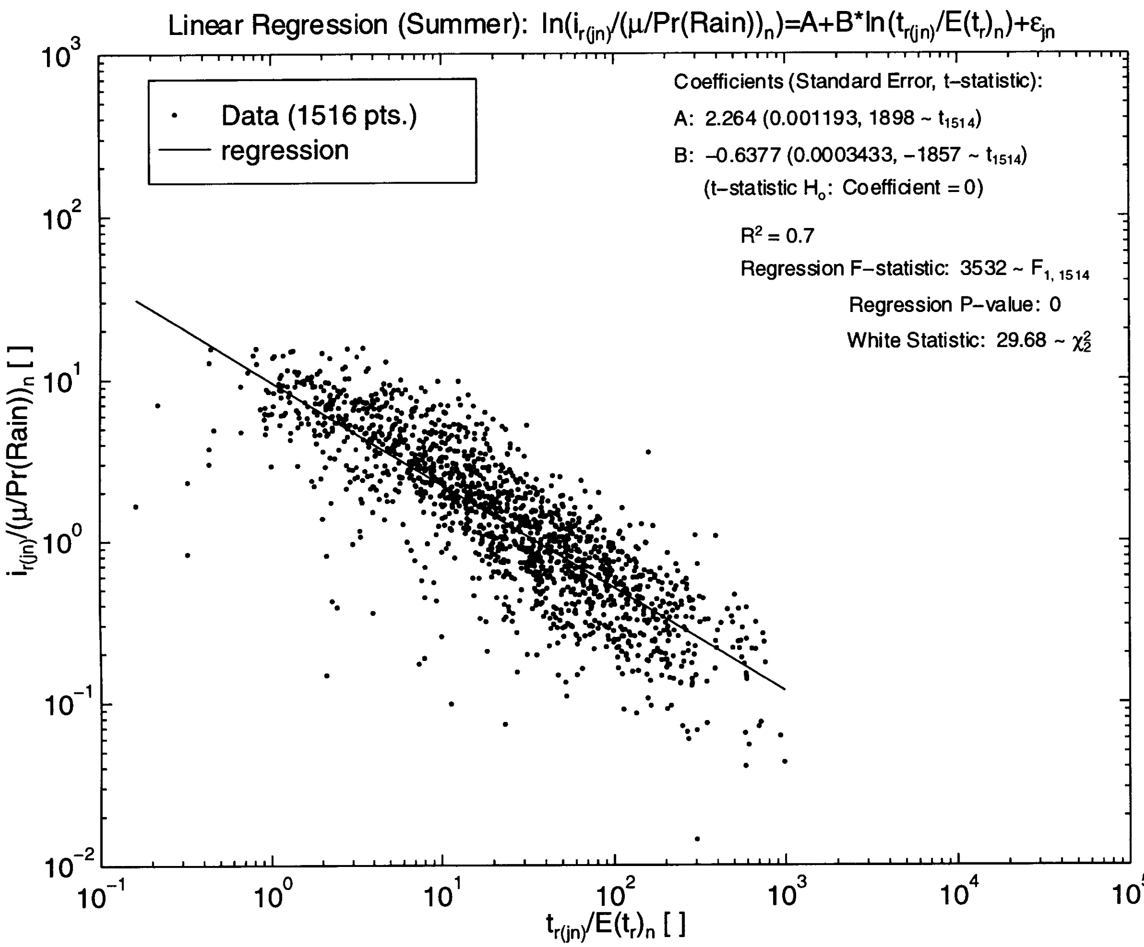

Figure C-6: Ordinary Least Squares Regression on equation C.2 with $I = \mu/Pr(Rain)$ and $T = E(t_r)$.

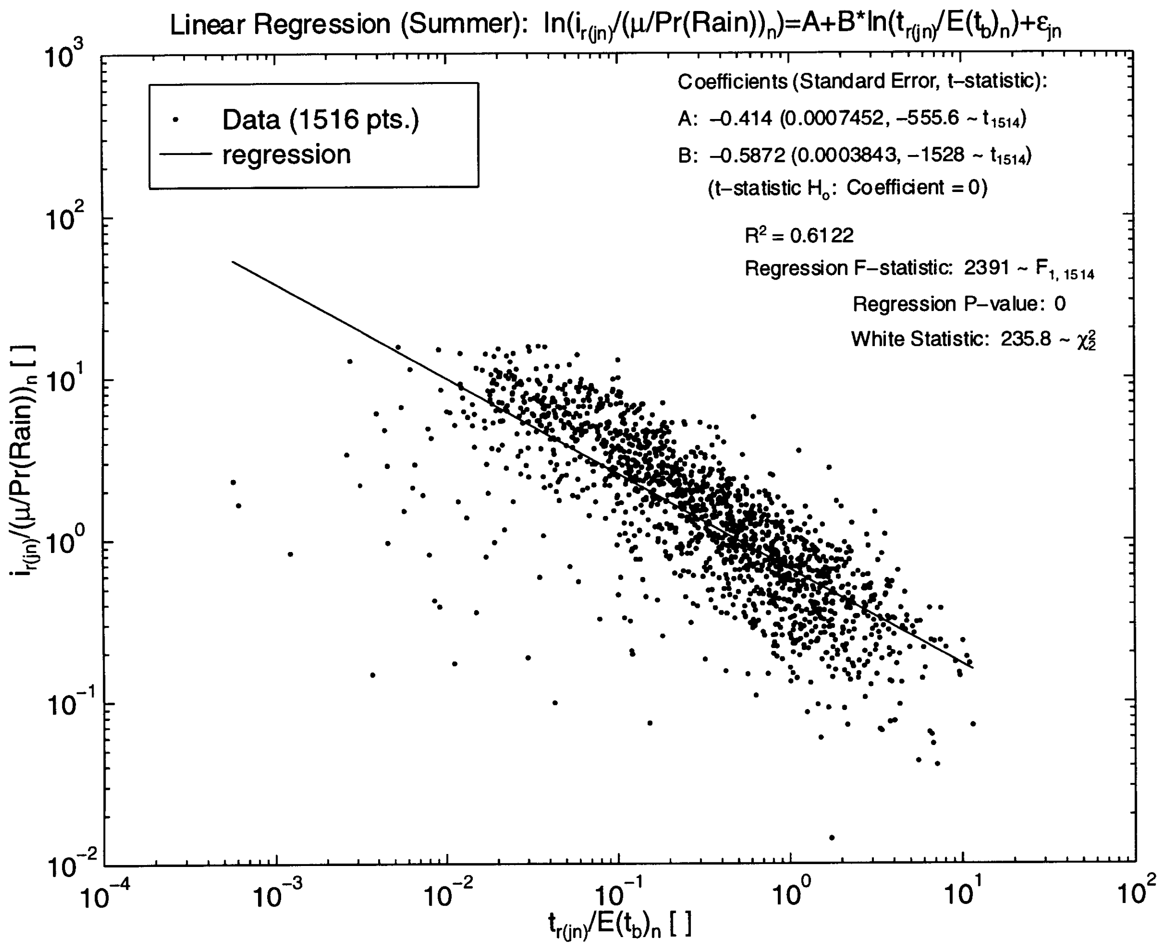

Figure C-7: Ordinary Least Squares Regression on Equation C.2 with $I = \mu/Pr(Rain)$ and $T = E(t_b)$.

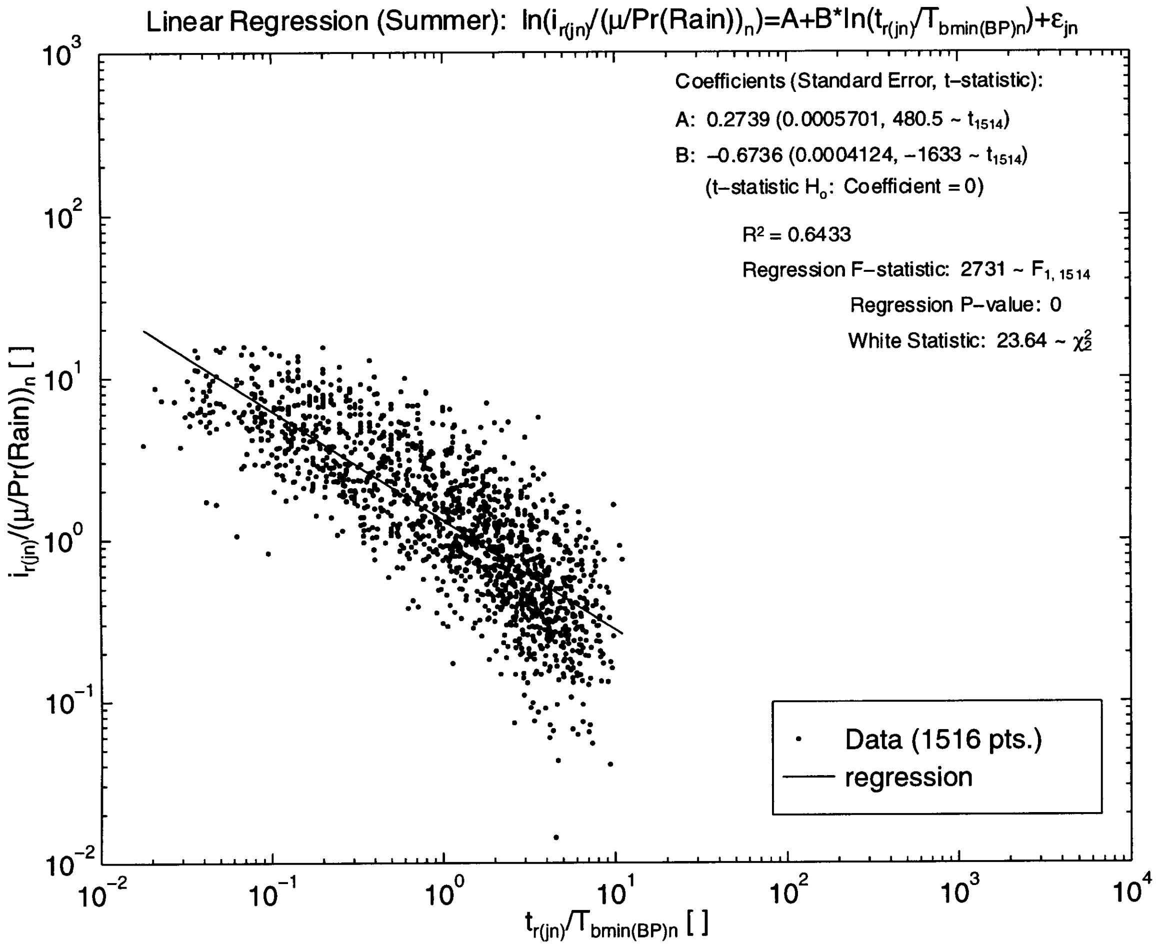

Figure C-8: Ordinary Least Squares Regression on equation C.2 with $I = \mu/Pr(Rain)$ and $T_{bmin(BP)}$.

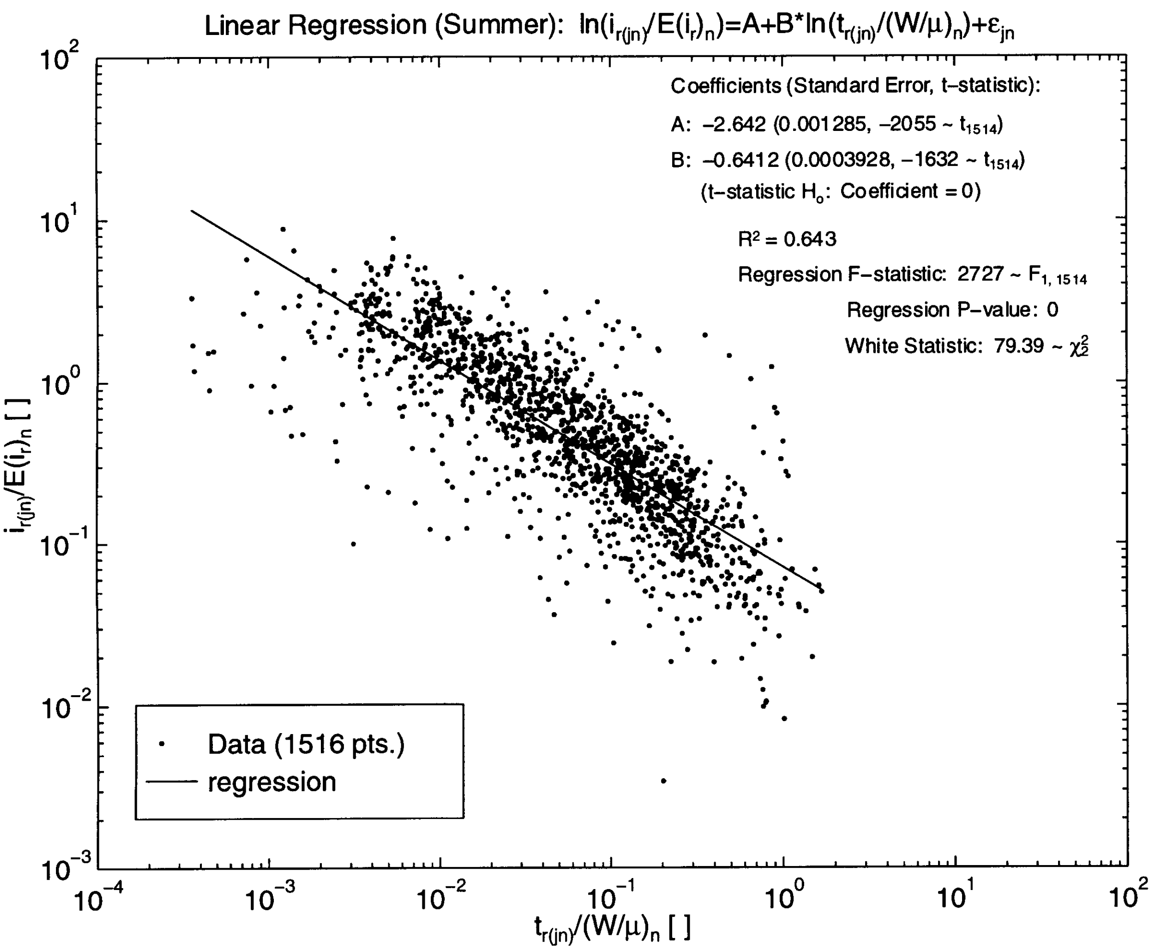

Figure C-9: Ordinary Least Squares Regression on Equation C.2 with $I = E(i_r)$ and $T = W/\mu$.

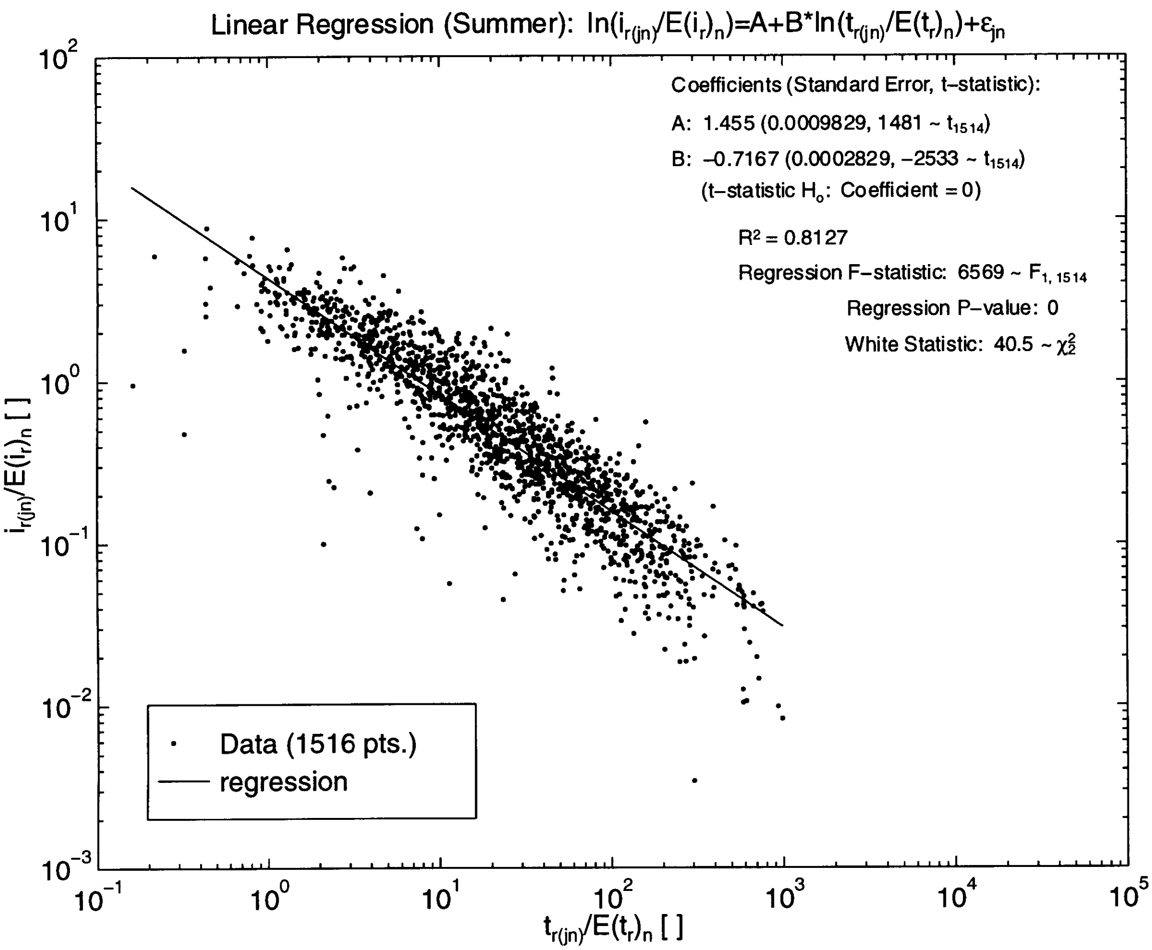

Figure C-10: Ordinary Least Squares Regression on Equation C.2 with $I = E(i_r)$ and $T = E(t_r)$.

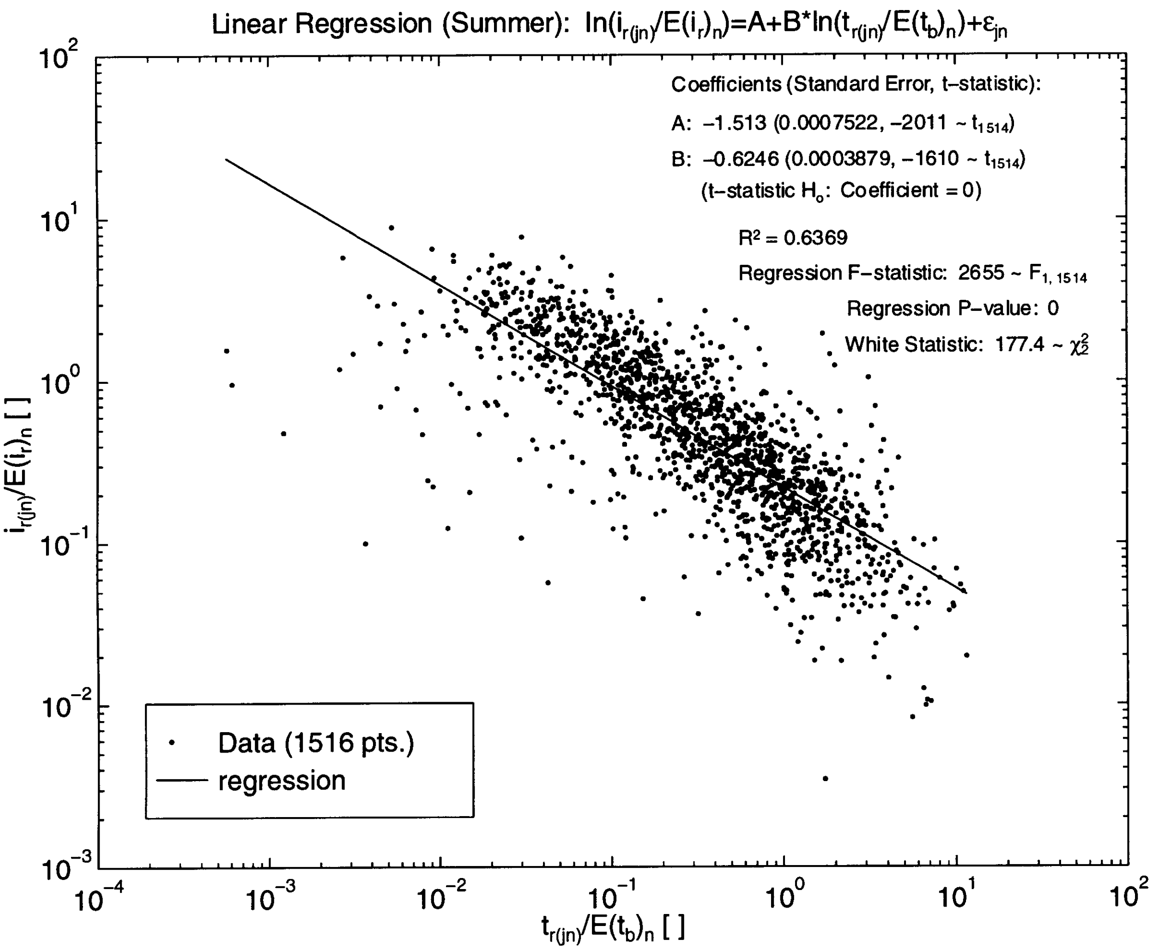

Figure C-11: Ordinary Least Squares Regression on Equation C.2 with $I = E(i_r)$ and $T = E(t_b)$.

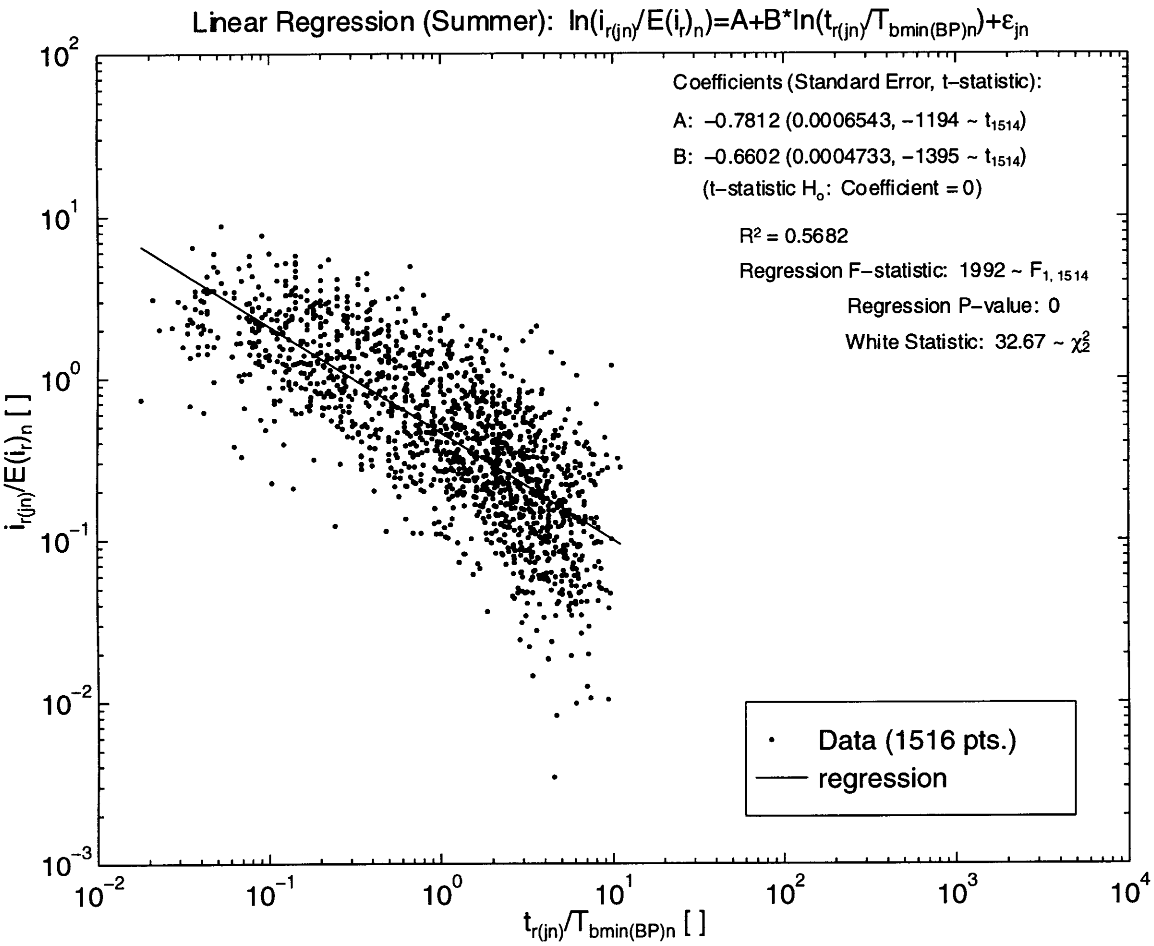

Figure C-12: Ordinary Least Squares Regression on Equation C.2 with $I = E(i_r)$ and $T = T_{bmin(BP)}$.

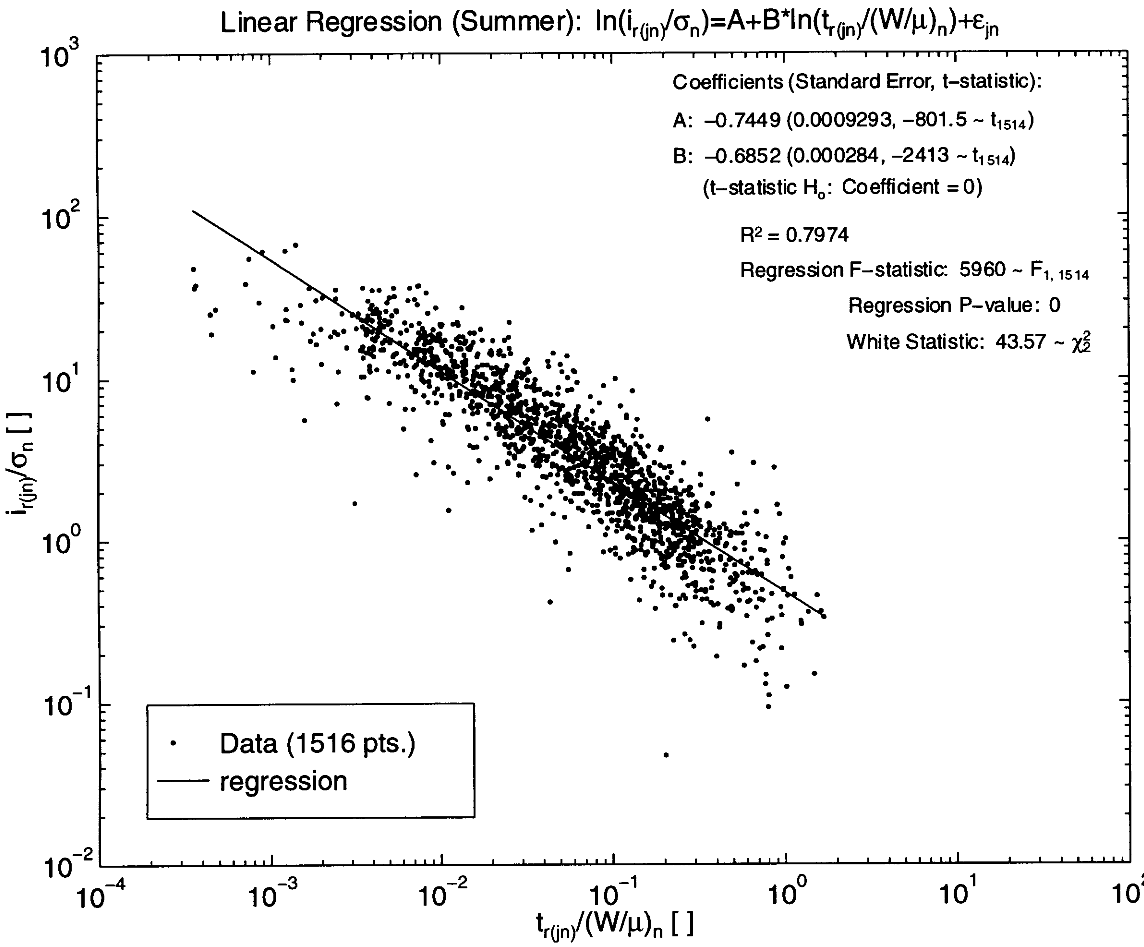

Figure C-13: Ordinary Least Squares Regression on Equation C.2 with $I = \sigma$ and $T = W/\mu$.

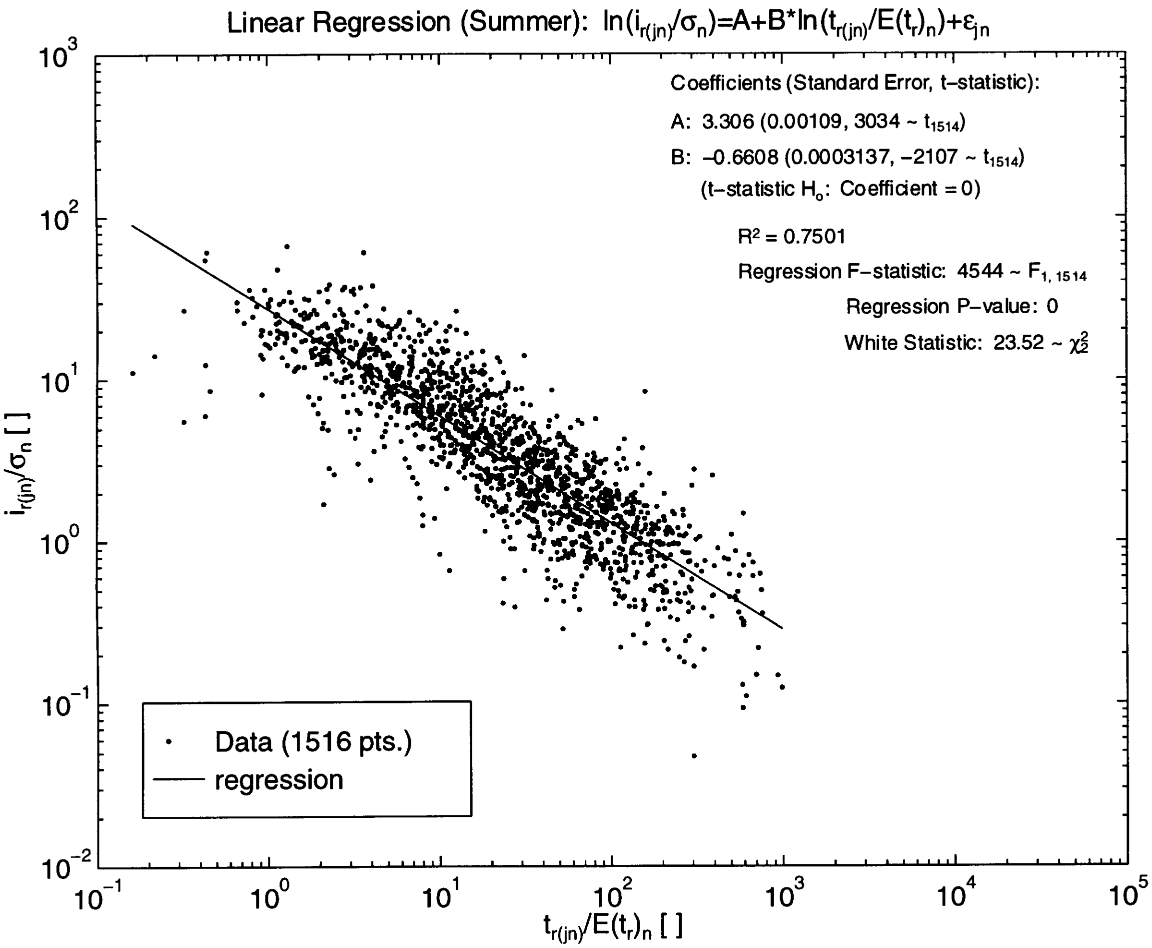

Figure C-14: Ordinary Least Squares Regression on Equation C.2 with $I = \sigma$ and $T = E(t_r)$.

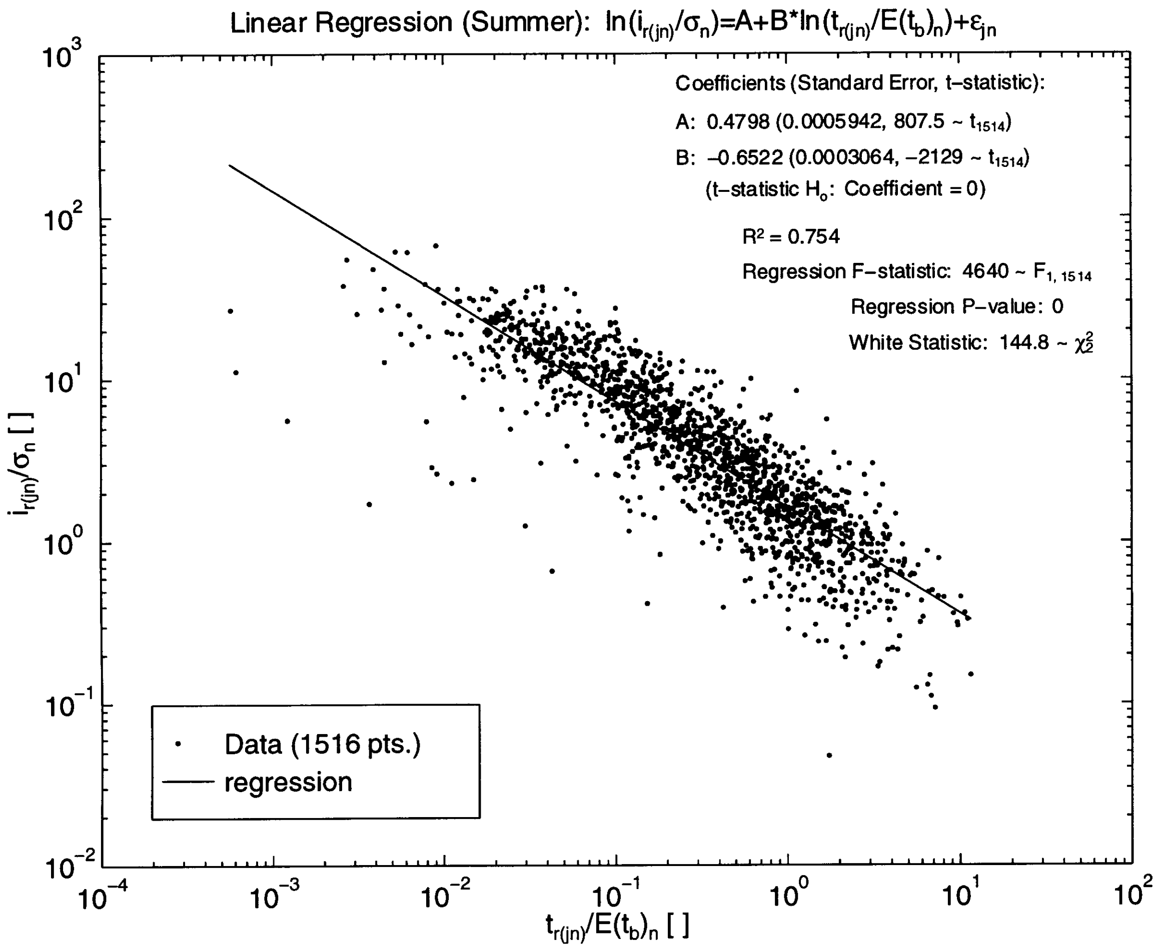

Figure C-15: Ordinary Least Squares Regression on Equation C.2 with $I = \sigma$ and $E(t_b)$.

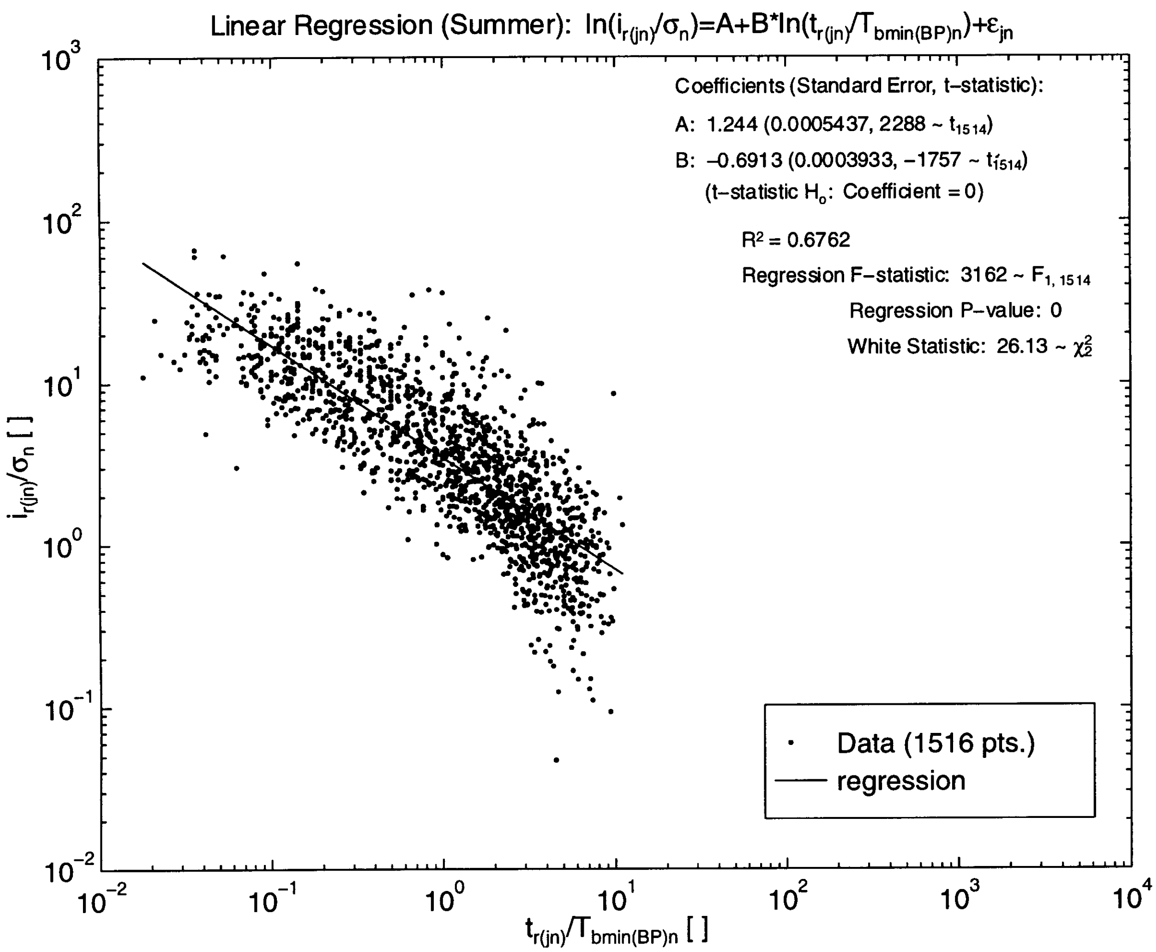

Figure C-16: Ordinary Least Squares Regression on Equation C.2 with $I = \sigma$ and $T = T_{bmin(BP)}$.

C.3 Heteroskedasticity and Statistical Tests

For fourteen out of the sixteen regressions, the White test for heteroskedasticty statistic (distributed χ_2^2) rejects the null hypothesis of homoskedasticity (constant variance of the error term) at the 95% level. Since the regression sample is large, we rely upon the asymptotic properties to yield consistant estimators, and to 'over-ride' small sample problems such as inefficiency. That is,

$$p\lim_{n\to\infty} \hat{\beta} = \beta \tag{C.3}$$

where β is the estimator of the regression coefficients. If this assumption holds, then we can assume that the estimates for the coefficients are reasonable and that the R^2 is close to its actual value. This conclusion is supported by the extremely low standard errors. These low standard errors yield very high t-statistics for the individual coefficients where the null hypothesis $H_\circ$ is that the coefficient is zero. The t-statistic for an individual coefficient is given as

$$\frac{\hat{\beta} - \beta_\circ}{s_{\hat{\beta}}} \sim t_{N-k} \tag{C.4}$$

where $\hat{\beta}$ is the estimated coefficient, $\beta_\circ$ is the null hypothesis value, N is the total number of data points, and k is the total number of regressors ($k = 2$, the constant and the variable). The 95% two-tailed level of significance value for the t distribution is 1.96. The t-statistics for all the regressed coefficients are at least two orders of magnitude higher than that. Therefore the null hypotheses of each coefficient equaling zero was rejected. Furthermore, the F-test statistic, which tests the null hypothesis that all the coefficients in a regression besides the regression constant are equal to zero,

$$\frac{R^2}{1 - R^2}\left(\frac{N-k}{k-1}\right) \sim F_{1,N-k} \tag{C.5}$$

is rejected at the 95% level for all of the regression. Again note that the statistics are, in general, four orders of magnitude grater than the 95% level F value. In fact, the P-values of the F-test are close to zero.

The implication of the White test's rejection of the null hypothesis of homoskedasticity is that the statistical tests, upon the regression and coefficient estimates (as described above), should be invalid due to the inefficiency introduced by the error term's non-constant variance. Considering the large sample, and the extreme values of the statistics, the inference one can draw from the test statistics (with the extreme values) is that the coefficients are probably significant. If the t-statistics were very close to the test value criteria at the chosen level of significance (in this case 95%), then the existance of heteroskedasticity would have to be weighed more carefully. To be cautious though, even with the large sample and sthe extreme statistical values, the occurance of heteroskedasticity should be stated.

Appendix D

Recovered Station PPF Curve Statistics

D.1 Overview

This appendix lists the statistics and accuracy measurements for the 141 stations which had greater than six GTBEFORE PPF points. The number of points is indicated in the '# Pts.' column. 'SER' is the station series Series I.D. number (SERID). A detailed discussion of the statistics follows in Section D.3. Section D.4 contains the coefficients obtained from the regressions of Equation 2.13 upon the individual stations.

D.2 Description of Table D.1 Data

RMSE is the root mean square error of the intensity, and lnBIAS is the bias of the natural log transformation of the intensity. A negative value for a lnBIAS value should be interpreted as a tendency for the recovered PPF to be above the GTBEFORE PPF points. The distance RMSE and the distance BIAS (DRMSE and DBIAS respectively) are calculated by computing the distances from a real PPF data (GTBEFORE) point to the closest point on the recovered PPF. A negative The recovered PPF point is determined by the point which, when connected to the real GTBEFORE data point, forms a line which is perpendicular to the slope of the recovered PPF at the recovered PPF point. Since the recovered PPF has a derivative which never changes sign, the perpendicular represents the minimum path from the real point to the curve. The one unfortunate drawback is that this minimum path technique is scale dependent.

Table D.1: Individual stations' recovered PPF statistics.

SER	# Pts.	RMSE	lnBIAS	$DRMSE_{i_r}$	$DBIAS_{i_r}$	$DRMSE_{t_r}$	$DBIAS_{t_r}$
1	11	2.9455	-0.0731	0.9775	0.0914	0.4452	-0.2109
2	7	3.5251	-0.2410	0.7968	-0.3594	0.8195	-0.2903
3	13	5.4241	-0.1205	0.2575	-0.1580	0.3242	-0.1015
4	11	2.9526	-0.1768	0.8924	-0.4108	0.4721	-0.0755
5	13	5.1871	-0.1452	0.6102	-0.2186	0.5467	-0.2134
6	17	4.0734	-0.2013	0.9504	-0.4226	0.9504	-0.5670
7	16	1.3021	-0.2745	0.6704	-0.2642	0.3521	0.0448
8	12	8.0439	-0.0351	1.0143	0.1449	1.1100	-0.6682
9	12	2.7345	0.0654	1.0169	0.3156	0.9872	0.5061
10	10	6.2042	0.2829	1.6031	0.8298	2.0209	1.2484
11	10	3.1516	-0.0292	0.7093	0.1116	1.2410	0.3357
12	10	2.9835	0.0276	0.4214	0.0746	0.3892	0.0493
13	16	2.2556	-0.3592	0.9990	-0.5095	0.8111	-0.2918
15	8	0.8786	-0.4900	0.7032	-0.3546	0.2886	-0.1793
16	11	1.7112	-0.0850	0.5448	-0.1328	0.5391	0.0844
17	9	1.8079	0.0458	0.6819	0.2770	0.3677	-0.0541
18	8	1.5277	-0.6120	0.9542	-0.5835	0.5832	-0.2831
23	10	0.4405	-0.0412	0.2014	0.0284	0.1785	0.0934
29	9	1.3508	0.1412	0.5693	0.2007	0.4246	0.2152
30	9	0.9809	-0.0244	0.3286	0.0218	0.4680	0.2073
31	9	2.1407	-0.3792	0.9150	-0.6325	0.6885	-0.4788
32	10	5.0137	0.1566	2.4683	1.0646	1.4472	0.5557
33	7	1.7000	0.1504	0.9576	0.5103	0.9575	0.4445
34	8	2.1380	0.2210	0.7688	0.4847	0.7104	0.3953
35	11	4.0980	-0.0352	0.2642	-0.0238	0.4132	-0.1062
36	11	3.2571	-0.0891	0.6925	-0.4314	0.5829	-0.0238
37	9	2.2864	0.0785	0.7823	0.2680	0.4291	0.1597
38	10	5.0608	-0.1704	0.9508	-0.5404	0.6831	-0.3747
39	9	6.0222	0.3280	1.1045	0.7397	0.8770	0.5333
40	13	0.8582	-0.2964	0.3589	-0.2088	0.3270	-0.0949
41	10	1.2917	-0.3821	0.7747	-0.6128	0.4847	-0.2981
42	7	0.6639	0.0026	0.4726	0.0760	0.2614	0.0084
43	11	2.7202	0.0064	0.6766	0.0647	0.4625	0.1347
45	10	4.4977	0.0861	1.9921	0.8767	1.0809	0.3617
47	10	3.0922	0.1365	0.5844	0.3511	1.4533	0.4709
48	9	7.2087	0.1911	1.3106	0.5287	1.5007	0.8103
49	11	2.0313	-0.2489	0.9541	-0.3314	0.3232	0.0788
50	12	1.9137	0.0868	0.7205	0.3322	0.3359	0.1042
51	12	3.3127	0.1308	1.1880	0.4610	1.2006	0.5621
52	12	4.4061	0.2542	0.7776	0.4934	1.0960	0.5920

SER	# Pts.	RMSE	lnBIAS	DRMSE$_{i_r}$	DBIAS$_{i_r}$	DRMSE$_{t_r}$	DBIAS$_{t_r}$
53	12	6.9524	0.2106	1.6084	0.6927	2.1613	0.9613
54	8	1.8399	0.1881	0.9023	0.6157	0.4350	0.3413
56	9	4.1852	0.0140	1.1890	0.1540	1.6745	0.8213
58	10	6.0752	0.2153	1.6726	0.9432	1.2987	0.9028
59	10	3.3251	-0.2333	0.9845	0.1536	1.2497	0.3411
60	7	2.8218	0.2858	1.7092	1.1329	1.2934	0.7039
61	13	2.9682	0.0874	0.6179	0.3735	0.7602	0.2426
62	7	4.1914	0.1780	1.4361	0.7728	0.5218	-0.0794
63	7	3.5450	0.1324	0.6110	0.2961	0.8351	0.5667
64	10	2.0448	0.1364	0.8379	0.5777	0.7274	0.4051
65	12	1.8052	-0.0130	1.4872	0.4666	0.4704	0.0382
66	7	4.8375	0.1857	0.5796	0.2915	1.1967	0.5440
68	7	1.2534	0.0946	0.8835	0.4584	0.5566	0.2633
69	9	1.8545	0.2985	1.0889	0.8409	0.6101	0.3688
71	9	5.0573	0.1603	0.5856	-0.0022	0.7094	0.0859
72	9	3.8981	-0.1290	0.4854	-0.1305	0.5680	-0.3125
74	13	1.6666	0.0081	0.6191	0.2522	0.3244	0.0270
75	14	2.4313	-0.0922	0.6636	-0.0211	0.2880	-0.0677
76	9	8.1248	0.1817	0.8551	0.2530	1.2143	0.6387
77	9	4.2483	0.1493	0.7971	0.4291	0.9033	0.4138
78	8	4.5051	0.1995	2.2960	0.9228	1.2322	0.7388
79	7	5.1625	0.1676	0.9027	0.4496	3.3835	1.4042
80	10	2.2273	0.1512	1.0370	0.5858	0.5303	0.2646
81	7	7.9301	0.2617	1.5433	0.3268	4.2881	2.4025
82	11	5.4745	-0.0819	1.1314	0.1940	0.7626	0.1713
83	12	1.2416	-0.0976	0.9208	-0.1618	0.4643	-0.1785
84	8	1.1152	0.0781	0.7534	0.2629	0.6016	0.2101
85	11	2.0770	0.0819	0.5938	0.1645	0.3659	0.0640
86	9	3.8576	0.1260	0.4207	0.1125	0.4743	0.1243
87	9	1.5571	-0.1123	0.2521	-0.1601	0.3159	0.0711
88	9	5.0732	0.0102	1.3024	0.3704	1.7672	0.8505
89	12	3.1700	0.1247	1.8092	0.5920	0.7700	0.3617
90	12	4.2399	0.0131	0.7079	0.0542	0.6003	0.1536
91	7	5.3369	0.4309	0.9581	0.6584	0.6104	0.3533
92	9	3.0290	0.1681	1.2678	0.5673	0.5582	0.2976
93	7	1.9555	-0.1831	0.6088	-0.3340	0.5701	-0.3501
94	11	0.7302	-0.5830	0.5237	-0.4273	0.2636	-0.1416
95	7	3.2376	0.2425	0.4132	0.0983	0.3047	0.0106
96	9	2.0137	0.0843	1.1188	0.4747	0.8328	0.4222
97	10	2.8526	0.0729	0.8801	0.3109	0.7322	-0.1818
100	8	8.1516	0.3481	0.8933	0.5197	1.7653	0.8013
101	10	3.9455	0.2441	1.3532	0.8612	1.4384	0.7550

SER	# Pts.	RMSE	lnBIAS	$DRMSE_{i_r}$	$DBIAS_{i_r}$	$DRMSE_{t_r}$	$DBIAS_{t_r}$
102	12	4.2072	-0.1976	0.9953	-0.4591	0.5231	-0.2904
103	9	1.8599	-0.0319	0.7035	-0.2682	0.5167	-0.1909
104	13	2.3609	-0.1498	1.3046	-0.0001	1.0788	0.3849
105	9	2.1116	0.0888	1.1011	0.4499	0.5850	-0.1455
106	11	3.5870	-0.0126	0.6851	-0.1499	0.9216	0.2677
107	13	3.9539	0.1630	1.1953	0.7675	0.4785	0.1370
108	12	2.8306	0.1456	0.5484	0.3459	0.6304	0.1598
110	8	2.5413	-0.1383	0.7662	-0.4261	0.7497	0.2297
111	8	2.2796	0.1093	1.1922	0.4360	0.7326	0.4225
112	11	1.4820	0.0975	0.6390	0.3244	0.5147	0.3432
113	10	5.8004	0.2370	1.3773	0.7015	0.7908	0.5403
114	8	2.8515	0.0941	2.1915	0.8896	1.1135	0.4956
115	10	1.3410	-0.1379	0.3063	-0.1049	0.1131	0.0125
116	14	1.1165	-0.2077	0.6315	-0.2966	0.4284	-0.1527
117	7	1.7136	-0.0720	0.5082	0.0100	0.5608	-0.1925
118	14	2.8072	-0.4466	0.6092	-0.3866	0.5951	-0.2358
119	11	0.4200	-0.2863	0.3525	-0.1069	0.1543	0.0218
120	10	2.5410	-0.4415	0.7931	-0.5870	0.6998	-0.3943
122	9	4.0726	0.1722	1.6497	1.0069	1.9325	0.7636
124	10	1.0388	0.0455	0.7028	0.2083	0.2175	0.0261
125	8	4.0519	0.2078	2.1183	1.1595	1.2705	0.7284
126	9	2.1292	-0.0188	0.5504	0.0705	0.2961	-0.0133
127	7	5.0385	0.4029	3.3424	2.1421	1.1487	0.5283
129	9	3.3046	0.0870	1.2318	0.3259	0.8169	-0.4754
130	11	4.9970	0.1782	0.9056	0.2590	0.8276	0.3360
131	9	0.4139	0.0143	0.3528	0.1802	0.1360	0.0704
132	11	4.5321	0.1675	1.9088	0.7674	1.1816	0.8489
133	11	6.5120	0.1335	0.6518	0.2721	0.9470	0.3972
134	17	2.2576	-0.1983	0.7648	-0.4825	0.4785	-0.2919
135	12	5.2086	-0.2293	0.8064	0.0765	0.8662	-0.0291
136	12	6.0397	-0.3292	1.5027	-0.9584	0.9662	-0.2691
137	15	3.6458	-0.1125	1.0012	-0.5111	0.5414	-0.2974
138	12	1.3906	-0.0451	0.6761	-0.0436	0.5084	-0.1814
139	11	1.9977	-0.0254	0.5693	0.1284	0.5341	0.2572
140	8	3.0080	0.2496	1.0374	0.4833	0.7143	-0.1533
141	9	5.7887	0.1507	2.0132	0.5925	0.9565	-0.5029
144	7	9.0755	0.4147	1.8525	1.1161	0.6804	0.1222
145	10	1.9131	0.2179	0.7851	0.4588	0.5012	0.1771
146	7	4.9284	0.1693	1.1938	0.6321	3.7091	1.3711
147	8	3.0422	0.0122	0.5751	0.2438	0.5270	0.1423
148	10	3.2569	0.1207	1.0214	0.5629	0.5443	0.3378
149	8	4.4967	0.1956	4.0734	1.7403	0.3489	0.1166

SER	# Pts.	RMSE	lnBIAS	DRMSE_{i_r}	DBIAS_{i_r}	DRMSE_{t_r}	DBIAS_{t_r}
150	10	1.5930	-0.0384	0.2916	0.0094	0.5581	0.1623
151	7	1.0030	0.0566	0.3984	-0.0101	0.4371	0.0695
155	11	1.7989	-0.0348	0.8680	0.1952	0.4054	0.1982
156	10	1.2202	-0.0311	0.5314	0.0611	0.3611	-0.0890
157	8	3.1576	0.1754	1.2531	0.7141	1.0801	0.5798
158	14	4.1436	0.0942	0.6898	0.3109	0.5479	0.0618
159	8	4.2186	0.0106	0.5992	-0.0205	0.4719	-0.2000
160	8	4.9595	-0.0386	1.2359	0.0967	0.7742	-0.3209
161	12	0.8914	-0.4273	0.6083	-0.3154	0.2929	0.0959
162	9	2.6200	-0.2200	0.5830	0.0220	0.5536	-0.1166
163	11	1.6870	-0.3617	0.9731	-0.6647	0.5668	-0.3526
164	7	1.2278	0.5554	1.2205	0.9486	0.0940	0.0522
165	10	4.2141	-0.4351	1.1419	-0.6719	1.0070	-0.4645
166	8	5.7951	-0.1365	1.3115	-0.3979	1.3597	-0.6755
168	11	3.3355	-0.0223	0.6306	-0.2327	0.6304	0.1428
169	9	1.6068	0.0080	0.5994	-0.1808	0.4806	-0.1592
170	15	1.9755	-0.2597	0.7964	-0.6068	0.4553	-0.1816

D.3 Discussion of Statistics

The first columns of Table D.2 are calculated with i_r as a function of t_r, where the t_r for the recovered PPF curve is matched to the $\tilde{t_r}$ of each of a stations GTBEFORE data point—the root mean square error (RMSE)

$$RMSE = \sqrt{\frac{1}{N_s}\left(\sum_{k=1}^{N_s}(\tilde{i_r} - i_r(t_r))^2\right)}, \tag{D.1}$$

and the natural log bias

$$lnBIAS = \frac{1}{N_s}\left(\sum_{k=1}^{N_s}(\ln\tilde{i_r} - \ln(i_r(t_r)))\right), \tag{D.2}$$

where N_s denotes the number of GTBEFORE data points of station s and $i_r(t_r)$ is derived from Equation 2.11 as

$$i_r(t_r) = \mu(\exp A)\left(\frac{t_r}{W/\mu}\right)^B. \tag{D.3}$$

The "~" denotes PPF points chosen by the GTBEFORE method, and a negative value of the natural log bias denotes a tendency of the recovered PPF to be above the GTBEFORE PPF points.

The problem with the standard RMSE (and the standard BIAS for that matter) is that, due to the hyperbolic nature of Equation C.1, it fails to capture how close the recovered curve is to the actual curve when t_r is close to zero,[1] When t_r is close to zero, $i_r(t_r)$ goes to infinity. So for low values of t_r, if the real data curve is shifted only slightly to one side of the recovered curve, the computed residual $i_r(t_r) - \tilde{i_r}$ can change dramatically. Thus the mean square error is misleading especially if data points are clustered around the low values of t_r. The high residuals imply a large difference between the curves and a poor fit, when in fact, around the extreme high and low regions of t_r the curves may visually seem very close together.

[1] Or when i_r is close to zero if one were computing the bias and root mean square error of t_r as a function of i_r instead.

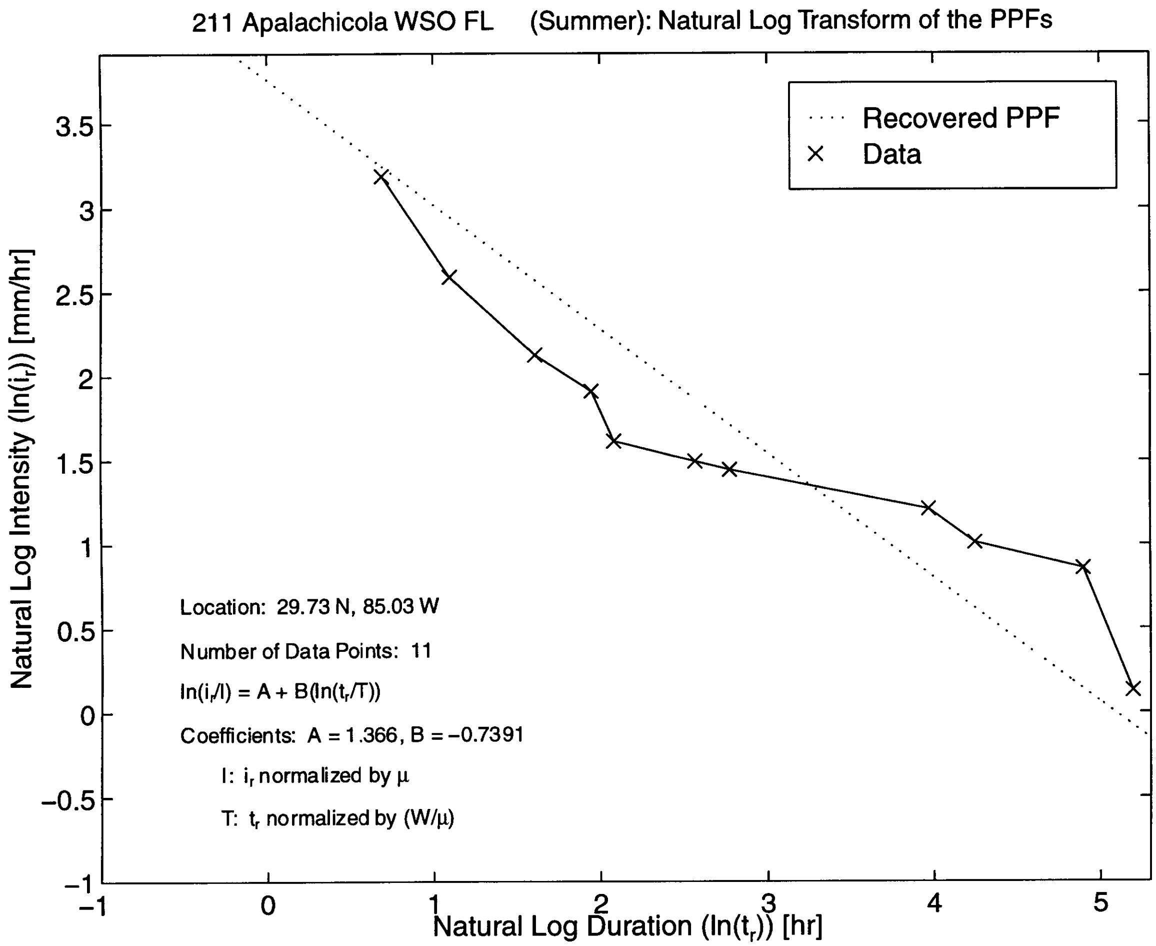

Figure D-1: Natural log-log transformation of the recovered and GTBEFORE linearly interpolated PPFs at Apalachicola, FL.

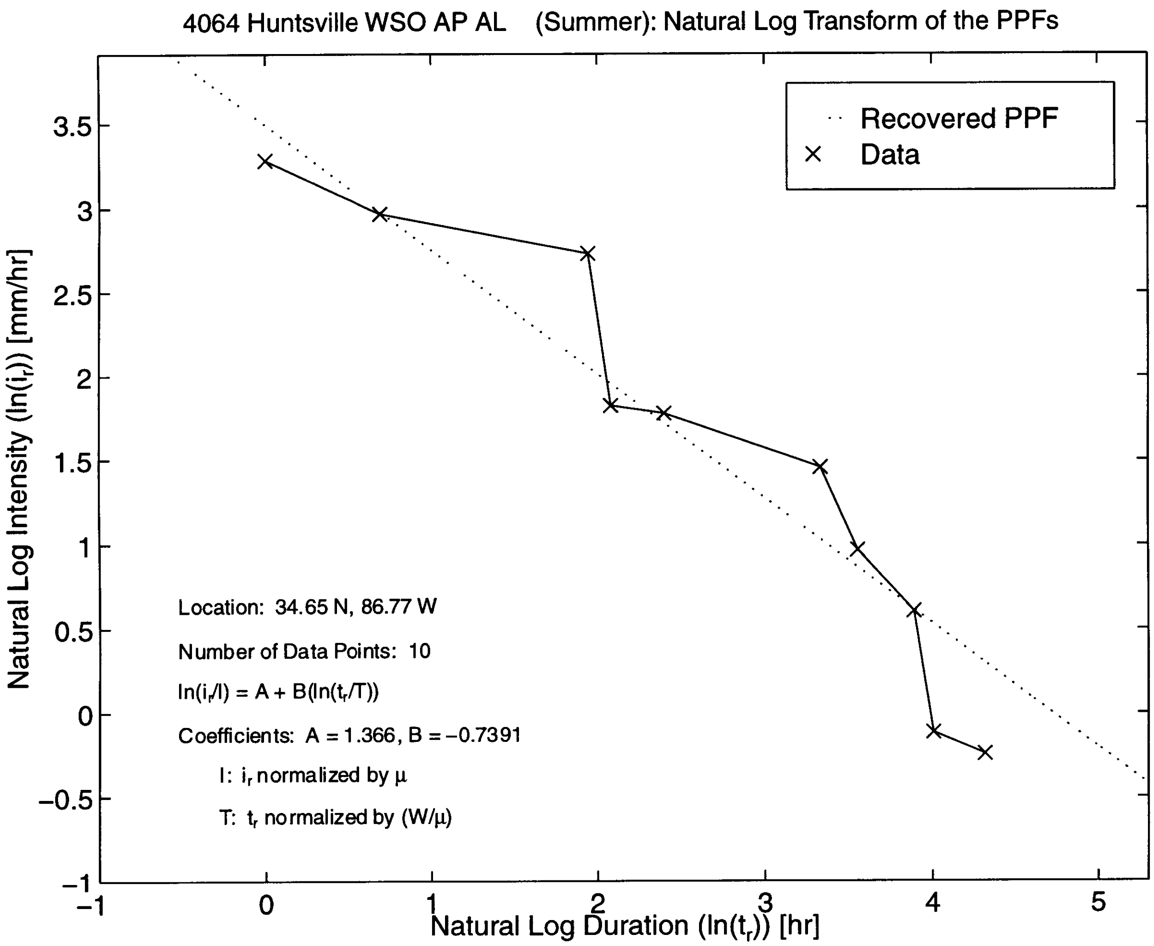

Figure D-2: Natural log-log transformation of the recovered and GTBEFORE linearly interpolated PPFs at Huntsville, AL.

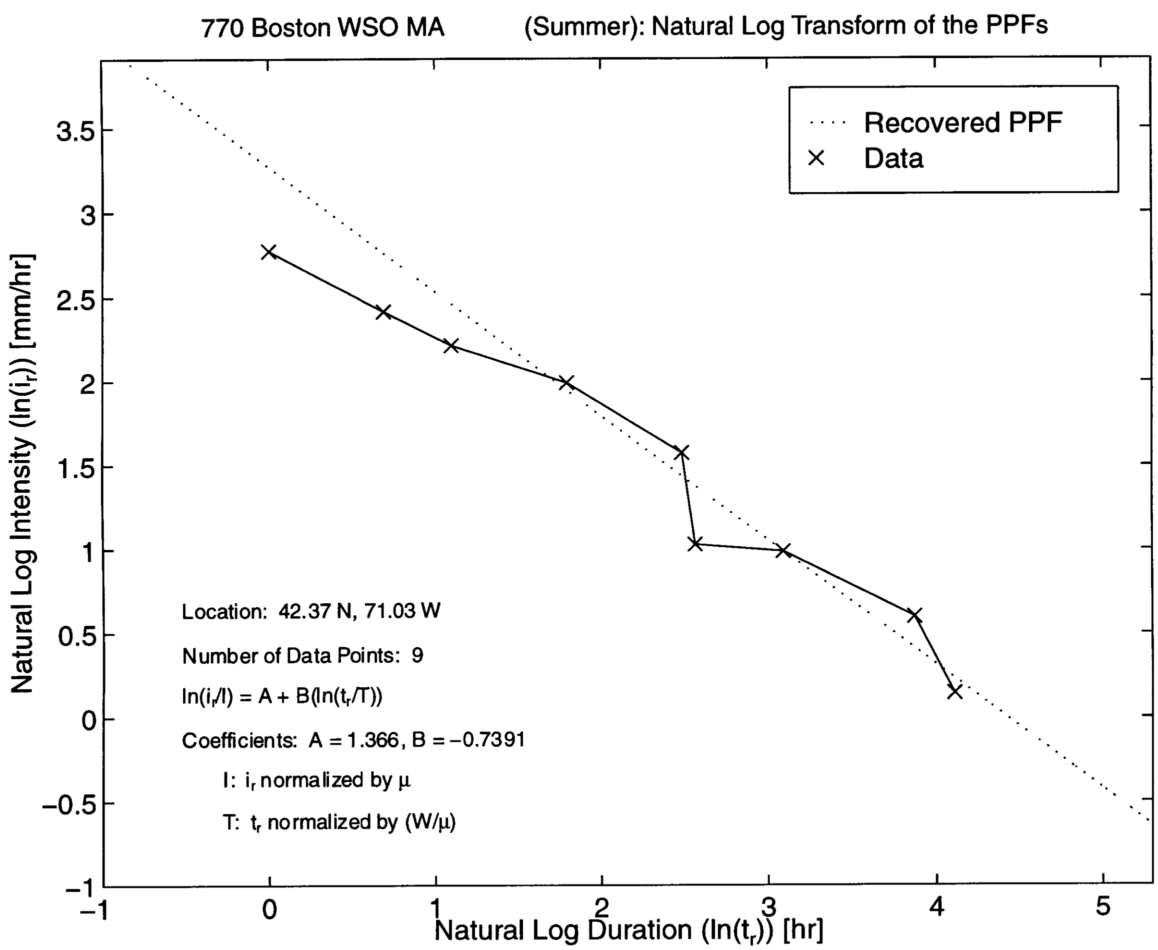

Figure D-3: Natural log-log transformation of the recovered and GTBEFORE linearly interpolated PPFs at Boston, MA.

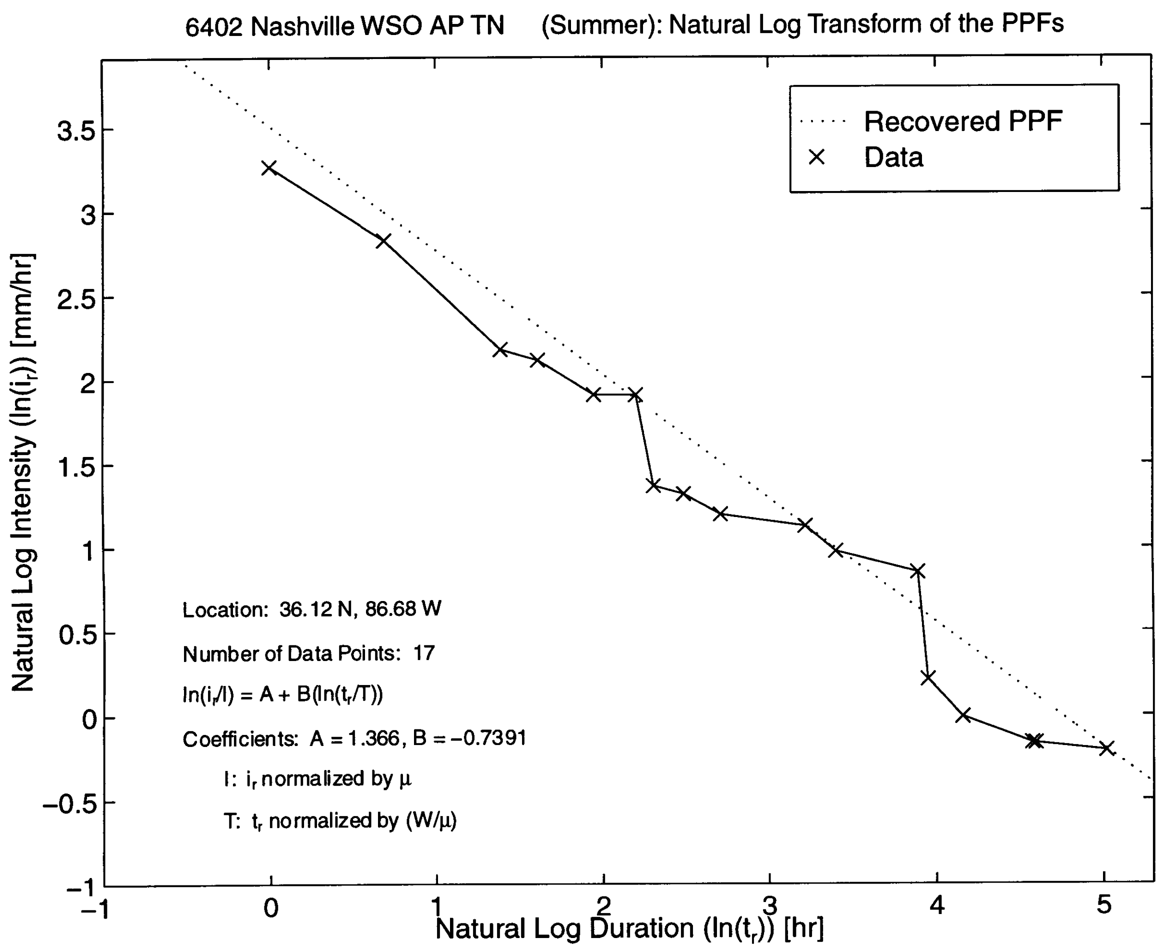

Figure D-4: Natural log-log transformation of the recovered and GTBEFORE linearly interpolated PPFs at Nashville,TN.

The natural log bias, which takes the difference of the natural logs of the i_r and $\tilde{i_r}$ values, attempts to deal with the hyperbolic problem by linearizing the recovered PPF and GTBEFORE points. Now the i_r errors at the extreme low values of t_r do not change geometrically or exponentially if the curves are offset, but rather, they change linearly (see figures D-1 through D-4). Unfortunately the natural log technique weights all of the differences between the GTBEFORE and the recovered PPFs evenly. Linearizing the two curves fails capture that at the extreme values of t_r all of the curves seem to be very similar and close together. It is in the mid-range of t_r that the curves visually look very different. Mathematically, in the mid-range is where individual hyperbolics functions can be very distinct from eachother. Therefore, the statistics should reflect the importance of the mid-range and the relative non-importance of the extremes.

D.3.1 Distance Statistics

Motivations

To reflect the importance of the mid-range and the more trivial differences of the extremes a new set of statistics were developed. These are the *Distance Statistics.* The significant departure from the statistics above is that these statistics are determined by the minimum distance from each GTBEFORE point to the recovered PPF. Since the recovered PPF function is monotonic, has a negative first derivative which never changes sign, and a positive second derivative which never changes sign,[2] the minimum distance from a point to the recovered PPF is where the line from the point to the PPF is prependicular to the rate of transformation at the place where the line contacts the PPF. Therefore the slope of the line is equal to the negative inverse of the rate of transformation at the point of contact. Figure D-5 demonstrates the concept graphically.

[2]The sign of the derivative is controlled by the sign of the coefficient B which, for the selected model, is negative. See Figure C-1.

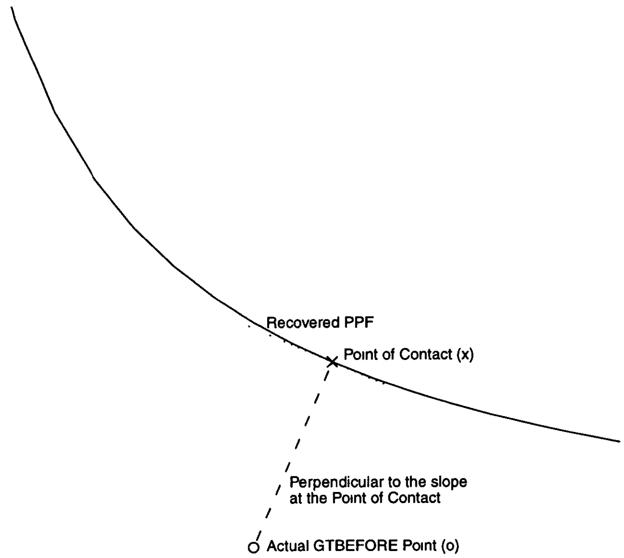

Figure D-5: The perpendicular line to the slope at the point of contact with the recovered PPF.

Theory

For clarity, the above is stated mathematically. The first derivative (with respect to t_r) of the functional form of the PPF (Equation C.1) is

$$\frac{di_r}{dt_r} = \frac{BI}{T}\exp(A)\left(\frac{t_r}{T}\right)^{B-1}, \tag{D.4}$$

the second derivative is

$$\frac{d^2 i_r}{{dt_r}^2} = \frac{(B-1)BI}{T^2}\exp(A)\left(\frac{t_r}{T}\right)^{B-2}, \tag{D.5}$$

and the condition for minimum distance from a GTBEFORE point to the recovered curve is when

$$-\left(\frac{di_r(t_r)}{dt_r}\right) = \frac{i_r(t_r) - \tilde{i_r}}{t_r - \tilde{t_r}} \tag{D.6}$$

where the right hand side of Equation D.6 represents the slope of the perpendicular line and the left hand side is the rate of transformation at the point of contact.

Substituting in Equation D.4 for di_r/dt_r and 2.11 for $i_r(t_r)$ into Equation D.6,

$$\frac{I\exp(A)\left(\frac{t_r}{T}\right)^B - \tilde{i_r}}{t_r - \tilde{t_r}} = -\left(\frac{BI}{T}\exp(A)\left(\frac{t_r}{T}\right)^{B-1}\right)^{-1} \tag{D.7}$$

is obtained. To develop Equation D.7 into a more solvable form, we extract a factor of negative one from the left hand side, multiply both sides by $(t_r - \tilde{t_r})$ to get t_r out of the denominator and then group the terms onto the left hand side. After some rearranging we have

$$\tilde{i_r} - I\exp(A)\left(\frac{t_r}{T}\right)^B + (\tilde{t_r} - t_r)\left(\frac{BI}{T}\exp(A)\left(\frac{t_r}{T}\right)^{B-1}\right)^{-1} = 0. \tag{D.8}$$

Equation D.8 is solved for t_r by using a Gauss-Newton method with a mixed quadratic and cubic line search procedure. To enable the Gauss-Newton method to work, it was important that t_r be taken out of of the denominator. Once t_r is found, i_r is obtained from Equation C.1. The i_r-t_r combination represents the point of the PPF that is the closest to the actual $\tilde{i_r}$-$\tilde{i_r}$ combination. The combinations are separated into intensity and duration components. The root means square error and bias are calculated in a similar fashion above, but this time, for each of the components. For example, the distance root mean square error in duration (DRMSE) for station s is

$$t_r\ DRMSE = \sqrt{\frac{1}{N_s}\left(\sum_{k=1}^{N_s}(\tilde{t_r} - t_r(t_r))^2\right)}, \tag{D.9}$$

and the distance bias of the intensity is

$$i_r\ DBIAS = \frac{1}{N_s}\left(\sum_{k=1}^{N_s}(\tilde{i_r} - i_r(t_r))\right) \tag{D.10}$$

where a negative $DBIAS$ implies that the recovered PPF is above the GTBEFORE points. The distance statistics can be seen here in Appendix D, Table D.1, and in Figures 3-3 through 3-6.

D.3.2 Comments on the Distance Statistics

On first examination Equations D.9 and D.10 look remarkably similar to Equations D.1 and D.2. The difference is restated again for arguments sake. In the regular statistics, the durations are matched. That is, t_r is equal to $\tilde{t_r}$. For the distance statistics, that constraint is relaxed. The distance statistics do not imply a direction of causality (duration⇒intensity) as may be implied in the standard statistics by matching the durations. The last sentence makes sense because in real life we do not know if one 'causes' the other. The distance statistics give a feeling of the the departure of the recovered curve from the actual data without being affected by the one axis effect of the extremes of the hyperbolic.

The one weakness of the distance statistics is that they are scale dependent. In order to achieve the visual perpendicularity, the whole number scales of each axis must be the same distance apart regardless of scale (i.e. the tick marks for mm/hr must be the same distance apart as that of the hour marks). One obtains different results if the axis units are varied. In this study, the axes are kept the same across stations. As the recovered PPFs become closer to the actual PPF points, the distance statistics tend to zero. Unlike the regular statistics, though, they will not be distorted by the extremes ends of the hyperbolic nature of the recovered PPF. The departures at the ends where the differences between the curves are small remain small.

Despite the introduction of the distance statistics, because uncertainty of any directional causality, and because the distance statistics are scale sensitive, the primary evaluation of the functional fit of the recovered PPF is still visual. From visual examination of Figures 3-3 through 3-6, the functional form does do reasonably well. It does approximate the hyperbolic nature of the GTBEFORE points and the general observed negative rate of transformation.

D.4 Individual Station Statistics

Tables D.2 and D.3 are the estimated coefficients from regressing Equation 2.13 upon individual stations. For Table D.1:

- A and B—Regression coefficients from Equation 2.13.
- $t_{A\circ}$ and $t_{B\circ}$—t-statistics for testing the null hypothesis that the coefficient (A or B respectively) is equal to zero.
- WT—White test statistic distributed χ^2_2
- DW—Durbin-Watson test statistic for first degree serial correlation (see *Pindyck and Rubinfeld* [22]. Like heteroskedasticity, serial correlation affects the efficiency of the estimators. If it is present, then the standard errors may not be correct, thereby leading to incorrect t-statistics.

For Table D.3, t_A and t_B represent the t-statistics for testing the null hypothesis that the respective coefficients A and B are equal to the general model values. C, from Equations 2.11 and 2.12, is equal to $\exp(A)$.

Table D.2: Statistics for individual station regressions.

SER	# Pts.	$A(=\ln C)$	B	$t_{A\circ}$	$t_{B\circ}$	WT	DW
1	11	1.8668	-0.5194	321.3636	-267.8950	2.8833	1.0597
2	7	0.9628	-0.8100	60.8104	-149.0830	5.0830	1.8659
3	13	1.3062	-0.6904	582.5437	-663.1093	1.8599	1.3902
4	11	1.0280	-0.8118	134.1721	-286.5653	4.8391	1.5038
5	13	1.2449	-0.7240	282.6408	-403.1621	3.1891	1.0866
6	17	1.2865	-0.6894	397.7641	-627.9023	2.6009	1.0319
7	16	0.7760	-0.9133	126.9139	-343.4860	6.8840	0.7700
8	12	1.8059	-0.5350	434.0311	-367.7697	7.3556	0.7670
9	12	0.8690	-0.9348	110.5056	-365.0668	0.3672	1.3816
10	10	0.7109	-1.0599	60.5053	-277.1312	7.1048	1.5328
11	10	1.1364	-0.8051	110.6795	-261.2730	0.1594	1.5221
12	10	1.6089	-0.6628	364.3163	-491.6239	2.4566	1.8172
13	16	1.2468	-0.6304	259.7576	-364.8514	13.2430	1.7343
15	8	-0.3848	-1.1068	-11.4124	-120.3245	5.9618	1.7614
16	11	0.8070	-0.8726	84.5068	-345.8948	3.4527	1.8698
17	9	1.5067	-0.7010	169.8662	-220.9982	2.6425	1.0531
18	8	1.5907	-0.5590	99.8818	-170.1087	0.2936	2.0740
23	10	0.8664	-0.8498	225.8376	-954.2374	0.9336	1.1354
29	9	1.5099	-0.7380	190.4365	-294.3220	2.5224	1.5894
30	9	1.0358	-0.8345	139.0242	-387.4819	4.4121	1.5045
31	9	1.5802	-0.6040	165.5878	-294.8505	2.8813	1.4117
32	10	1.6100	-0.7072	102.7551	-135.2779	0.5122	0.5296
33	7	1.5690	-0.7223	176.5098	-268.5462	1.7640	2.1361
34	8	1.8824	-0.6474	237.1855	-275.9388	2.8683	1.5164
35	11	1.6812	-0.6203	560.4892	-672.3783	1.2917	2.0368
36	11	1.1811	-0.7803	224.3834	-384.9043	10.3605	1.7451
37	9	1.4134	-0.7528	230.4962	-357.3238	1.5090	2.1417
38	10	1.5169	-0.6401	210.7807	-317.0497	1.9387	1.5766
39	9	1.6129	-0.7743	356.1819	-472.2316	1.5902	1.7794
40	13	0.8608	-0.8068	73.0708	-222.7654	6.3773	0.7785
41	10	1.2253	-0.6765	199.4653	-440.8588	1.1027	2.1415
42	7	1.1569	-0.7961	61.2554	-164.7902	0.7793	1.2833
43	11	1.0463	-0.8676	123.2318	-290.2067	1.6659	1.0875
45	10	1.3845	-0.7692	147.9553	-220.3615	0.6665	0.9864
47	10	1.4587	-0.7543	154.4810	-253.3915	0.1616	1.8863
48	9	0.5934	-1.1292	52.9931	-270.3914	2.0068	1.6229
49	11	0.1878	-1.0496	17.6639	-312.8478	1.6739	1.7638
50	12	1.3502	-0.7716	197.0099	-367.8406	0.7141	0.8761
51	12	1.1963	-0.8385	148.7410	-331.6544	6.0510	1.1796
52	12	1.3042	-0.8529	195.8960	-380.0033	0.5681	1.2465

SER	# Pts.	$A(=\ln C)$	B	$t_{A\circ}$	$t_{B\circ}$	WT	DW
53	12	0.9365	-0.9769	102.1739	-317.2808	0.5391	0.9170
54	8	1.4833	-0.7622	152.6015	-252.9260	4.8012	2.4401
56	9	-0.7665	-1.4647	-29.5944	-170.8517	4.2351	2.0236
58	10	0.6232	-1.0505	38.3908	-211.3354	9.2076	1.3948
59	10	0.6386	-0.9150	47.8079	-215.3133	0.3646	0.6505
60	7	1.7603	-0.7044	116.4982	-154.9213	5.7614	2.3250
61	13	1.3865	-0.7705	185.2337	-279.2738	8.2066	0.8065
62	7	2.3731	-0.4908	374.8282	-282.5707	2.7781	2.7327
63	7	0.9625	-0.9256	117.6349	-343.9967	2.4069	3.1894
64	10	1.1952	-0.8481	132.8212	-284.3523	7.3962	1.4896
65	12	1.1705	-0.8170	124.3077	-238.0890	4.8965	0.6522
66	7	1.3693	-0.8355	232.2211	-342.5087	1.6138	2.6258
68	7	1.2143	-0.8251	129.2886	-264.6089	0.1414	1.9236
69	9	2.1488	-0.5932	1012.7985	-1001.1632	4.9654	3.5725
71	9	1.7409	-0.6567	153.2214	-163.6063	4.7518	1.1226
72	9	1.6468	-0.6194	293.9782	-406.5037	2.3301	2.0363
74	13	1.4154	-0.7236	332.6363	-495.7922	0.2789	0.9026
75	14	1.2733	-0.7392	208.6859	-379.9993	0.0197	1.0682
76	9	0.8341	-0.9692	64.3033	-244.7832	1.3170	1.3972
77	9	1.3752	-0.7819	112.5781	-226.1921	2.2533	1.5324
78	8	0.3190	-1.2636	19.7473	-196.1445	0.3401	2.0447
79	7	0.7855	-1.0551	41.7883	-147.5674	0.2337	2.4526
80	10	1.2101	-0.8813	166.4308	-300.7335	0.3093	1.4665
81	7	1.0346	-0.9249	32.7361	-99.6317	1.1463	1.2847
82	11	1.2900	-0.7368	167.6963	-297.9440	2.7486	1.4882
83	12	1.2094	-0.7627	205.4655	-367.8085	2.7069	1.2281
84	8	1.3307	-0.7791	188.7832	-353.6235	3.7076	2.6095
85	11	1.1943	-0.8245	128.2942	-284.3979	0.5598	0.9423
86	9	1.0057	-0.9367	75.8762	-193.0438	0.2056	1.3373
87	9	0.9280	-0.8557	149.2133	-421.2459	1.8430	1.8266
88	9	0.0661	-1.3356	5.0531	-245.9743	1.1070	2.0748
89	12	0.8399	-0.9947	90.4817	-290.4989	0.8014	1.1947
90	12	1.0406	-0.8738	107.2097	-261.6803	1.6268	0.9548
91	7	1.4706	-0.8572	177.5039	-315.4482	0.2865	2.8678
92	9	1.2687	-0.8384	131.8637	-272.3982	1.0161	2.4933
93	7	1.9948	-0.5396	205.5703	-244.4645	3.6975	3.2612
94	11	0.0980	-0.8931	6.3066	-264.6324	3.6037	0.7731
95	7	1.5929	-0.7432	121.9585	-230.3324	0.7925	2.3614
96	9	1.1792	-0.8233	93.7536	-226.1069	2.2289	1.5407
97	10	1.7559	-0.5586	249.2721	-171.9599	5.4656	1.3399
100	8	1.0993	-0.9597	83.0987	-215.6957	2.6058	3.1260
101	10	1.6302	-0.7325	137.2630	-203.9236	3.0383	0.7362

SER	# Pts.	$A (= \ln C)$	B	$t_{A\circ}$	$t_{B\circ}$	WT	DW
102	12	1.4071	-0.6472	245.3425	-342.3606	1.8016	1.8322
103	9	1.4408	-0.7003	241.6701	-372.1926	1.0709	1.6430
104	13	0.8248	-0.9072	122.0806	-353.7316	5.6590	1.3073
105	9	1.9250	-0.5880	193.1544	-196.4922	8.4930	2.4119
106	11	1.1975	-0.7905	159.2604	-347.0617	1.8666	2.0377
107	13	1.6393	-0.6833	285.5432	-292.5803	3.2293	1.1807
108	12	1.7729	-0.6461	483.1131	-564.8387	1.3617	2.2264
110	8	0.2507	-1.0165	18.1395	-266.8726	0.2781	2.1246
111	8	1.1372	-0.8584	113.9810	-256.2873	1.4116	1.8593
112	11	1.2463	-0.8088	180.1803	-378.9511	3.1643	3.1538
113	10	1.0700	-0.9214	133.5614	-354.8685	3.1710	1.5904
114	8	0.7645	-1.0073	57.9530	-211.9559	1.4509	2.1818
115	10	0.6172	-0.8959	85.7341	-508.0648	1.2084	1.9549
116	14	1.3119	-0.6663	240.9077	-291.2093	1.1831	0.3694
117	7	1.5371	-0.6651	81.2393	-122.5675	2.5625	1.1314
118	14	0.8768	-0.7536	103.6736	-284.2845	0.4712	0.3438
119	11	0.1936	-1.0359	18.7092	-314.5454	2.0658	0.6241
120	10	1.4298	-0.6202	382.1551	-754.0223	1.5102	2.1273
122	9	1.6232	-0.7096	112.2139	-153.6788	1.1315	1.0042
124	10	1.4872	-0.7139	174.8686	-268.1555	3.5957	1.9181
125	8	1.3222	-0.8191	55.2254	-116.2280	3.1167	0.9582
126	9	1.4851	-0.6951	289.2054	-461.4751	2.9960	2.6555
127	7	2.7573	-0.4445	228.8589	-132.6053	1.1284	1.1115
129	9	1.7992	-0.5822	251.6317	-228.7672	3.1618	1.5928
130	11	1.4438	-0.7794	186.0190	-288.9111	0.0811	1.6347
131	9	1.2357	-0.7951	119.4631	-228.8896	7.5800	2.2194
132	11	0.7836	-0.9699	58.0428	-247.2883	0.0235	1.3268
133	11	0.9548	-1.0085	155.3664	-398.7492	2.7400	2.0511
134	17	1.2060	-0.7251	364.7704	-678.2499	3.2428	1.6477
135	12	1.0010	-0.7894	98.5117	-246.2266	0.8785	0.7692
136	12	0.5265	-0.9198	35.2982	-189.9502	2.6116	0.7804
137	15	1.4078	-0.6794	270.2774	-381.1766	6.1675	1.3703
138	12	1.3738	-0.7194	344.7818	-557.5975	4.1157	1.9420
139	11	0.7851	-0.9242	171.2142	-656.7068	0.9823	1.9261
140	8	2.1588	-0.5601	286.7213	-251.4451	6.4645	1.8468
141	9	2.2206	-0.4863	261.5481	-186.0211	1.2570	2.1732
144	7	2.2089	-0.5980	176.6651	-158.2847	3.3679	1.5338
145	10	1.4464	-0.8079	189.1615	-239.2930	3.6517	1.2439
146	7	1.4526	-0.7697	79.7701	-139.4117	1.4136	1.8767
147	8	1.3125	-0.7596	109.2068	-232.7594	2.0369	1.0121
148	10	1.0908	-0.8583	128.3873	-363.2435	0.9590	1.6801
149	8	1.5218	-0.7594	104.0031	-140.7348	0.8328	1.0246

SER	# Pts.	$A(=\ln C)$	B	$t_{A\circ}$	$t_{B\circ}$	WT	DW
150	10	0.8829	-0.8719	63.9935	-232.9023	2.6359	1.2823
151	7	1.4936	-0.7205	141.9988	-283.8920	2.1184	3.0946
155	11	0.8842	-0.8978	77.4445	-238.9438	3.9191	1.4845
156	10	1.3109	-0.7488	148.5644	-238.7257	7.5956	1.9001
157	8	1.3456	-0.8009	152.5265	-306.1394	1.1972	1.2905
158	14	1.5983	-0.6920	328.3006	-458.7067	6.7410	2.1697
159	8	1.8746	-0.5735	358.8643	-375.3015	2.6453	1.8075
160	8	1.9216	-0.5670	122.4107	-132.6076	5.5260	2.1394
161	12	-0.9342	-1.2626	-62.1120	-306.9175	1.8052	0.8167
162	9	1.0495	-0.7644	53.3490	-155.1679	4.2505	1.3051
163	11	1.2738	-0.6482	194.3861	-327.7742	1.2048	1.3258
164	7	1.8804	-0.7803	222.2697	-125.7071	3.2610	1.0253
165	10	1.5448	-0.5326	164.4198	-191.0446	1.7158	0.5891
166	8	1.7878	-0.4275	237.6126	-150.0712	1.0716	1.0480
168	11	0.9870	-0.8601	152.7997	-430.5351	1.4408	2.0551
169	9	1.6883	-0.6337	244.0829	-288.9911	3.1987	1.6448
170	15	0.9582	-0.7978	217.9296	-505.2130	2.4906	1.6343
171	7	2.0495	-0.6245	181.2216	-198.1000	0.5941	2.1108
172	8	1.0825	-0.8107	87.7164	-233.6464	2.4376	1.1556
173	9	1.2335	-0.8124	133.2144	-253.6479	0.7899	2.0243

Table D.3: Statistics for individual station regressions (continued).

SER	# Pts.	$A(=\ln C)$	B	C	t_A	t_B
1	11	1.8668	-0.5194	6.4677	86.2648	113.3479
2	7	0.9628	-0.8100	2.6190	-25.4479	-13.0514
3	13	1.3062	-0.6904	3.6920	-26.5504	46.7298
4	11	1.0280	-0.8118	2.7953	-44.0847	-25.6607
5	13	1.2449	-0.7240	3.4725	-27.4379	8.4322
6	17	1.2865	-0.6894	3.6200	-24.4941	45.2378
7	16	0.7760	-0.9133	2.1728	-96.4359	-65.5051
8	12	1.8059	-0.5350	6.0854	105.7937	140.3093
9	12	0.8690	-0.9348	2.3845	-63.1680	-76.4192
10	10	0.7109	-1.0599	2.0359	-55.7268	-83.8896
11	10	1.1364	-0.8051	3.1156	-22.3294	-21.4197
12	10	1.6089	-0.6628	4.9972	55.0632	56.5892
13	16	1.2468	-0.6304	3.4792	-24.7732	62.8786
15	8	-0.3848	-1.1068	0.6806	-51.9209	-39.9717
16	11	0.8070	-0.8726	2.2411	-58.5086	-52.9336
17	9	1.5067	-0.7010	4.5118	15.8947	12.0145
18	8	1.5907	-0.5590	4.9070	14.1254	54.7958
23	10	0.8664	-0.8498	2.3784	-130.1382	-124.3222
29	9	1.5099	-0.7380	4.5265	18.1910	0.4202
30	9	1.0358	-0.8345	2.8173	-44.2837	-44.3006
31	9	1.5802	-0.6040	4.8558	22.4732	65.9219
32	10	1.6100	-0.7072	5.0029	15.5921	6.1024
33	7	1.5690	-0.7223	4.8020	22.8727	6.2455
34	8	1.8824	-0.6474	6.5693	65.1053	39.0853
35	11	1.6812	-0.6203	5.3722	105.1935	128.7095
36	11	1.1811	-0.7803	3.2580	-35.0681	-20.3492
37	9	1.4134	-0.7528	4.1097	7.7716	-6.5158
38	10	1.5169	-0.6401	4.5580	21.0072	49.0301
39	9	1.6129	-0.7743	5.0172	54.5845	-21.4928
40	13	0.8608	-0.8068	2.3650	-42.8641	-18.6957
41	10	1.2253	-0.6765	3.4052	-22.8537	40.7877
42	7	1.1569	-0.7961	3.1801	-11.0555	-11.7993
43	11	1.0463	-0.8676	2.8471	-37.6211	-42.9726
45	10	1.3845	-0.7692	3.9929	2.0096	-8.6346
47	10	1.4587	-0.7543	4.3006	9.8525	-5.1196
48	9	0.5934	-1.1292	1.8102	-68.9663	-93.4142
49	11	0.1878	-1.0496	1.2066	-110.8064	-92.5424
50	12	1.3502	-0.7716	3.8584	-2.2563	-15.4976
51	12	1.1963	-0.8385	3.3078	-21.0647	-39.3317
52	12	1.3042	-0.8529	3.6846	-9.2436	-50.7056

SER	# Pts.	$A(=\ln C)$	B	C	t_A	t_B
53	12	0.9365	-0.9769	2.5510	-46.8287	-77.2359
54	8	1.4833	-0.7622	4.4076	12.1017	-7.6730
56	9	-0.7665	-1.4647	0.4646	-82.3230	-84.6389
58	10	0.6232	-1.0505	1.8648	-45.7467	-62.6500
59	10	0.6386	-0.9150	1.8938	-54.4385	-41.3931
60	7	1.7603	-0.7044	5.8144	26.1162	7.6313
61	13	1.3865	-0.7705	4.0007	2.7732	-11.3872
62	7	2.3731	-0.4908	10.7304	159.1145	142.9290
63	7	0.9625	-0.9256	2.6183	-49.2744	-69.3199
64	10	1.1952	-0.8481	3.3041	-18.9534	-36.5531
65	12	1.1705	-0.8170	3.2237	-20.7282	-22.7034
66	7	1.3693	-0.8355	3.9327	0.6154	-39.5131
68	7	1.2143	-0.8251	3.3680	-16.1182	-27.5801
69	9	2.1488	-0.5932	8.5742	369.0830	246.1961
71	9	1.7409	-0.6567	5.7026	33.0229	20.5294
72	9	1.6468	-0.6194	5.1903	50.1775	78.5395
74	13	1.4154	-0.7236	4.1180	11.6692	10.6506
75	14	1.2733	-0.7392	3.5727	-15.1411	-0.0399
76	9	0.8341	-0.9692	2.3029	-40.9772	-58.1186
77	9	1.3752	-0.7819	3.9558	0.7751	-12.3710
78	8	0.3190	-1.2636	1.3758	-64.7945	-81.4185
79	7	0.7855	-1.0551	2.1935	-30.8663	-44.1988
80	10	1.2101	-0.8813	3.3539	-21.3979	-48.5205
81	7	1.0346	-0.9249	2.8140	-10.4764	-20.0122
82	11	1.2900	-0.7368	3.6328	-9.8430	0.9122
83	12	1.2094	-0.7627	3.3514	-26.5606	-11.4022
84	8	1.3307	-0.7791	3.7836	-4.9704	-18.1754
85	11	1.1943	-0.8245	3.3014	-18.4070	-29.4489
86	9	1.0057	-0.9367	2.7338	-27.1613	-40.7224
87	9	0.9280	-0.8557	2.5295	-70.3785	-57.4078
88	9	0.0661	-1.3356	1.0683	-99.3771	-109.8612
89	12	0.8399	-0.9947	2.3162	-56.6432	-74.6400
90	12	1.0406	-0.8738	2.8309	-33.4937	-40.3375
91	7	1.4706	-0.8572	4.3518	12.6601	-43.4571
92	9	1.2687	-0.8384	3.5561	-10.0870	-32.2530
93	7	1.9948	-0.5396	7.3508	64.8305	90.3960
94	11	0.0980	-0.8931	1.1030	-81.5399	-45.6333
95	7	1.5929	-0.7432	4.9181	17.3967	-1.2835
96	9	1.1792	-0.8233	3.2519	-14.8258	-23.1166
97	10	1.7559	-0.5586	5.7885	55.3903	55.5790
100	8	1.0993	-0.9597	3.0020	-20.1392	-49.5882
101	10	1.6302	-0.7325	5.1048	22.2691	1.8482

SER	# Pts.	$A(=\ln C)$	B	C	t_A	t_B
102	12	1.4071	-0.6472	4.0843	7.2242	48.6302
103	9	1.4408	-0.7003	4.2240	12.5938	20.6433
104	13	0.8248	-0.9072	2.2813	-80.0727	-65.5577
105	9	1.9250	-0.5880	6.8549	56.1169	50.5053
106	11	1.1975	-0.7905	3.3118	-22.3720	-22.5548
107	13	1.6393	-0.6833	5.1516	47.6578	23.8869
108	12	1.7729	-0.6461	5.8881	110.9674	81.3319
110	8	0.2507	-1.0165	1.2849	-80.6780	-72.8292
111	8	1.1372	-0.8584	3.1182	-22.8981	-35.6267
112	11	1.2463	-0.8088	3.4775	-17.2592	-32.6486
113	10	1.0700	-0.9214	2.9155	-36.9041	-70.2153
114	8	0.7645	-1.0073	2.1479	-45.5733	-56.4353
115	10	0.6172	-0.8959	1.8537	-103.9863	-88.9435
116	14	1.3119	-0.6663	3.7132	-9.8805	31.8224
117	7	1.5371	-0.6651	4.6511	9.0588	13.6323
118	14	0.8768	-0.7536	2.4032	-57.8059	-5.4841
119	11	0.1936	-1.0359	1.2137	-113.2393	-90.1286
120	10	1.4298	-0.6202	4.1780	17.1387	144.5810
122	9	1.6232	-0.7096	5.0694	17.8020	6.3766
124	10	1.4872	-0.7139	4.4248	14.2875	9.4797
125	8	1.3222	-0.8191	3.7517	-1.8163	-11.3471
126	9	1.4851	-0.6951	4.4153	23.2470	29.2147
127	7	2.7573	-0.4445	15.7576	115.5046	87.8690
129	9	1.7992	-0.5822	6.0447	60.6260	61.6462
130	11	1.4438	-0.7794	4.2369	10.0639	-14.9242
131	9	1.2357	-0.7951	3.4407	-12.5725	-16.1260
132	11	0.7836	-0.9699	2.1894	-43.1142	-58.8397
133	11	0.9548	-1.0085	2.5981	-66.8703	-106.5095
134	17	1.2060	-0.7251	3.3403	-48.2897	13.1123
135	12	1.0010	-0.7894	2.7209	-35.8949	-15.7031
136	12	0.5265	-0.9198	1.6929	-56.2701	-37.3248
137	15	1.4078	-0.6794	4.0868	8.0725	33.4795
138	12	1.3738	-0.7194	3.9504	2.0331	15.3006
139	11	0.7851	-0.9242	2.1927	-126.6121	-131.5192
140	8	2.1588	-0.5601	8.6609	105.3359	80.3595
141	9	2.2206	-0.4863	9.2126	100.6897	96.6713
144	7	2.2089	-0.5980	9.1056	67.4369	37.3574
145	10	1.4464	-0.8079	4.2476	10.5471	-20.3920
146	7	1.4526	-0.7697	4.2743	4.7731	-5.5385
147	8	1.3125	-0.7596	3.7153	-4.4306	-6.2937
148	10	1.0908	-0.8583	2.9768	-32.3500	-50.4353
149	8	1.5218	-0.7594	4.5803	10.6657	-3.7611

SER	# Pts.	$A(=\ln C)$	B	C	t_A	t_B
150	10	0.8829	-0.8719	2.4180	-34.9913	-35.4729
151	7	1.4936	-0.7205	4.4529	12.1551	7.3361
155	11	0.8842	-0.8978	2.4209	-42.1798	-42.2336
156	10	1.3109	-0.7488	3.7095	-6.2106	-3.0891
157	8	1.3456	-0.8009	3.8404	-2.2811	-23.6344
158	14	1.5983	-0.6920	4.9447	47.7783	31.2506
159	8	1.8746	-0.5735	6.5181	97.4172	108.3598
160	8	1.9216	-0.5670	6.8317	35.4102	40.2634
161	12	-0.9342	-1.2626	0.3929	-152.9131	-127.2613
162	9	1.0495	-0.7644	2.8562	-16.0735	-5.1312
163	11	1.2738	-0.6482	3.5744	-14.0257	45.9773
164	7	1.8804	-0.7803	6.5560	60.8373	-6.6423
165	10	1.5448	-0.5326	4.6872	19.0654	74.0501
166	8	1.7878	-0.4275	5.9763	56.0988	109.4032
168	11	0.9870	-0.8601	2.6832	-58.6287	-60.5651
169	9	1.6883	-0.6337	5.4103	46.6377	48.0562
170	15	0.9582	-0.7978	2.6070	-92.6847	-37.1862
171	7	2.0495	-0.6245	7.7640	60.4624	36.3592
172	8	1.0825	-0.8107	2.9522	-22.9443	-20.6450
173	9	1.2335	-0.8124	3.4334	-14.2736	-22.8886

Appendix E

Storm Determination

The following is the description and code of the program used to find independent storms from precipitation timelines.

IRTR.f and supporting Subroutines

Program Written by Conan L. Hom

Conceptualization Date: October 1996

Completion Date: August, 1997

Revision for S.M. thesis Date: Thursday, October 22, 1997

Language: Fortran77 PDQ

Supporting Subroutines in *irtrsubs.f*

File Units used: 10, 44, 51, 52, 54, 55, 56, 57, 63, 70, 71, 72, 73, 74, 80, 82, 90, 91.

E.1 General Description

IRTR.f takes the precipitation data of the National Climate Data Center in the format used in Jennifer K. Wynn's thesis [33], and applies the Breakpoint and/or Coefficient of Variation determined T_{bmin} to break down the precipiation timelines into series of independent storms. In addition, it finds the precipitable water and the various hourly rainfall statistics for the given raingauge stations analyzed. The precipitable water data is given on a grid which has a resolution of one degree [13]. The seasonal precipitable water is determined by the monthly averages of 1988 through 1992. For stations which are not exactly on grid points, the precipitable water is linearly interpolated between grid points. This becomes more important as one progresses towards the equator where the 1 x 1 degree areas become larger. The key subroutine for the selection of independent storms is HTRCOUNT.

IRTR.f and its supporting subroutines select storms from seasonal timelines: Winter (December–February), Spring (March–May), Summer (June–August), and Autumn (September–November). One potential problem is that a storm may overlap between seasons. HTRCOUNT truncates storms at the end of a time period. *IRTR.f* does have the capacity to assign storms of the same depth one i_r-t_r combination (see subroutine STORMCDF—*irtrsubs.f*, l.1027).

This program also can fill in missing data time lines with zeros, skip trying to count storms from seasons with missing data, or throw out files with missing data. For the purposes of this thesis, the data was filtered twice. First, files with major amounts of missing data (more than two days) were thrown out and then, in the remaining files, an individual station years' seasons with missing data were skipped.

E.2 Input

IRTR.f needs the following files and parameter inputs (inside INPUT.dir):

- Rain gauge files (series#.txt).
- filelistnol.dat—Series no, Station ID no, Lat, Long, Station Location.
- scatfall.dat—BP and CV T_{bmin} data for fall.
- scatspr.dat–BP and CV T_{bmin} data for spring.
- scatsum.dat– BP and CV T_{bmin} data for summer.
- scatwin.dat— BP and CV T_{bmin} data for winter.
- series#.tst files: The rainfall data is in 1/100 inch units. The output is converted into mm/hr (1/100 inch = 0.254 mm).

- parameter NF must be set to the number of stations to be processed.
- contents in PW.dir, specifically the PWAvg_fall.dat (sprg, summ, and wint too).

E.3 Output

IRTR.f outputs (inside OUTPUT.dir):

- stations–list of stations used in the current run and their locations.
- filenames–junk file
- sprgBP#.out, summBP#.out, fallBP#.out, wintBP#.out:
  - # is the series I.D. (SERID) number.
  - CV indicates that the Coefficient of Variation t_{bmin} was used.
  - BP indicates that the Breakpoint t_{bmin} was used.
  - The columns are:
    1. Cumulative Distribution Value of the storm depth.[1]
    2. Storm Depth [mm].
    3. Intensity [mm/hr].
    4. Intensity without 0 precipitation periods averaged in.
    5. Duration [hr].
    6. Duration without 0 precipiation periods counted.
- stormcount - number of storms in the files. The columns are:
  1. Series I.D. number.
  2. station I.D. number.
  3. T_{bmin} type (1 =BP, 2 =CV).

[1]If storms have the same depth, the order of appearance in the cumulative distribution is arbitrary. As stated before, the program can be made to assign same depth storms one i_r-t_r combination.

4. No. of Storms–Spring, Summer, Autumn, Winter.

- statPW.dat–station precipitable water (W) data [mm].
- statAVRG.dat–station average hourly precipitation (μ) [mm/hr].
- statVARN.dat–station hourly precipitation variance (σ^2) [(mm/hr)^2].
- statLAG1.dat–station lag one (hourly) auto correlation ($\rho(1)$) [ ]. See Equation B.1.

E.4 Programs

E.4.1 Main Program: IRTR.f

```
C234567
      program IRTR
*********|*********|*********|*********|*********|*********|*********|**
*     Written by Conan L. Hom
*     MM/DD/YY
*     Massachusetts Institute of Technology
*********|*********|*********|*********|*********|*********|*********|**
*     See IRTRreadme for variable descriptions
*********|*********|*********|*********|*********|*********|*********|**
*     Also uses files irtrsubs.f and splines.f
*********|*********|*********|*********|*********|*********|*********|**
      parameter (NF=161, NM = 12, NY=15, N =20000,NS = 900, ISE = 4)
      parameter (NSTAT = 174, NH = 24, ZERO = 1E-5)
      parameter (NORM = 0, NDATA = 2)

*     Ndata = 1 if fill in missing data with zeros (exludes early ending
*             files (in sercheck (line 244))
*     Ndata = 2 eliminate seasons with missing data
*        (for statistics seasonstat line 353)
*        (for stormcounting seasonstorm)
*     Ndata = 0 throw out files with missing data.


***********MATRICES CREATION*******************************************
*   SCRATCHFILE
      integer SFILE(NSTAT,3), ISCRATCH
*   FILELOAD
      character*23 LABEL(NSTAT)
      integer IDSER(NSTAT),IDSTAT(NSTAT)
```

```
      real LAT(NSTAT), LONG(NSTAT)
*     NAMEIN
      character*30 FNAME
      integer NOFILE(NSTAT),IS
*     STATIONFIND
      integer ISTAT
      real LATSTAT, LONGSTAT
      character*23 LABSTAT
*     PWENTER
      real PW(ISE),PWALL(NSTAT,ISE+2)
      character seasons(ISE)*4
*     SERCHECK
      integer II(N,3), NYEAR(NM), LYEAR(NM), MONTHS(NM)
      real STID(3), C(N,NH)
*     SEASONSTAT
      real D(ISE,4),P(ISE),S(ISE),SCOUNT(ISE),ST(ISE),S2(ISE),S3(ISE)
      real S4(ISE)
*     STATSMAKE
      real AVG(NSTAT,ISE+2),VARI(NSTAT,ISE+2),PRNO(NSTAT,ISE+2)
      real LAG1(NSTAT,ISE+2)
*     TBMINLOAD
      integer ITB,TBSTAT(NSTAT)
      real TB(NSTAT,2),MINTB(ISE)
*     NAMEOUT
      character*30 NAM(ISE)
      integer ITB
*     SEASONSTORM
      real DEPTH(NS,ISE), IR(NS,ISE),IRNZ(NS,ISE),TR(NS,ISE),
     $     TRNZ(NS,ISE)
      integer ISTART(ISE),X(N)
*     HTRCOUNT
      real H(NS), STR(NS), STRNZ(NS)
      integer ISTR(ISE)
*     STORMCDF
      real CDF(NS,ISE)
*     STORMNORM
***********END MATRICES CREATION*****************************************

      data NYEAR /31,28,31,30,31,30,31,31,30,31,30,31/
      data LYEAR /31,29,31,30,31,30,31,31,30,31,30,31/
      data seasons /'sprg','summ','fall','wint'/
      open (unit=80, file='OUTPUT.dir/stormcount', status='unknown')
      open (unit=82, file='OUTPUT.dir/stations', status = 'unknown')
      open (unit=44, file='Teststat', status = 'unknown')
      write(82,*) 'SERNO STID  LAT      LONG    STATION'

      icrud = 0
      do 100 i = 1,NF
        is = i
        call FILELOAD(IDSER,IDSTAT,LABEL,LAT,LONG,NSTAT)
        call NAMEIN(FNAME,IS,NF,NOFILE,NSTAT)
        call STATIONFIND(IDSER,IDSTAT,IS,ISTAT,LABEL,LABSTAT,LAT,
     $       LATSTAT,LONG,LONGSTAT,NOFILE,NSTAT)
        call PWENTER(LATSTAT,LONGSTAT, PW,ISE,SEASONS)
```

```
      call PWALLMAKE(PW,ISE,PWALL,NSTAT,IS,NOFILE,ISTAT)

101    continue
      call SERCHECK(C,FNAME,II,IND,LYEAR,MONTHS,N,NYEAR,NM,NH,NY,
   $       STID,ZERO,ISCRATCH,NDATA)
      write(*,*) 'POST SCRATCH'
      call SCRATCHFILE(ISCRATCH,SFILE,NSTAT,ISTAT,NOFILE,IS,ICRUD)
      if(ISCRATCH.gt.0.and.NDATA.eq.0) goto 100
      if(ISCRATCH.eq.2) goto 100
      call SEASONSTAT(C,D,IND,ISE,LYEAR,MONTHS,N,NH,NM,NY,NYEAR,P,
   $     S,SCOUNT,ST,S2,S3,S4,ZERO,ISTAT,NOFILE,NSTAT,NDATA,IS)
      call STATSMAKE(D,ISE,AVG,VARI,PRNO,LAG1,NSTAT,IS,NOFILE,ISTAT)
*      works up to this point!!!

      do 200 ITB =1,2
        call TBMINLOAD(ISE,ISTAT,ITB,MINTB,NSTAT,TB,TBSTAT)
            write(*,*) 'TBMIN ',(MINTB(jj),jj = 1,ISE)
        call NAMEOUT(IS,ITB,ISE,NAM,NF,NOFILE,NSTAT)
            write(*,*) (NAM(jj),jj=1,ISE)
        call SEASONSTORM(C,DEPTH,H,IR,IRNZ,ISE,ISTART,ISTR,LYEAR,
   $         MINTB, N, NH, NM, NS, NY, NYEAR, STR, STRNZ, TR, TRNZ,
   $         X,ZERO,NDATA)

        if (NORM.eq.1) then
          call STORMNORM(DEPTH,IR,IRNZ,ISE,ISTR,NS,P,PW,TR,TRNZ)
        endif

        call STORMCDF(CDF,DEPTH,IR,IRNZ,ISE,ISTR,NS,TR,TRNZ,ZERO)

        call MAKEOUT(CDF,DEPTH,IR,IRNZ,ISE,ISTR,NAM,NS,TR,TRNZ)

        write(80,199) NOFILE(i), ISTAT,ITB, (ISTR(k),k=1,4)
199       format(1x,I4,1x,I5,1x,I2,1X,4(I4,1x))
200     continue

100  continue
    call PWPRINT(ISE, PWALL, NF,NSTAT)
    call STATSPRINT(ISE,AVG,VARI,PRNO,LAG1,NF,NSTAT)
    call SCRATCHPRINT(SFILE,NSTAT,ICRUD)
    close(unit = 80, status = 'keep')
    close(unit = 82, status = 'keep')
    close(unit = 44, status = 'keep')
    END
```

E.4.2 Supporting Subroutines: irtrsubs.f

```
          subroutines for irtr.f

*********|*********|*********|*********|*********|*********|*********|**
*               SUPPORTING SUBROUTINES
*********|*********|*********|*********|*********|*********|*********|**

      SUBROUTINE ABOVEBELOW(X,Xhi,Xlow)
*********|*********|*********|*********|*********|*********|*********|**
*     Conan L. Hom
*********|*********|*********|*********|*********|*********|*********|**
      real X, Xhi, Xlow

      Xhi = real(int(X)) +1.
      Xlow = real(int(X))
      RETURN
      END



      subroutine FILELOAD(IDSER,IDSTAT,LABEL,LAT,LONG,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*     Loads in the series numbers with the corresponding Stations.
*     IDSTAT (list of all the station numbers)
*     IDSER  (list of all the series numbers)
*        IDSTAT(i) corresponds with IDSER(i) for all i.
*     Conan L. Hom
*     Saturday, November 30, 1996
*********|*********|*********|*********|*********|*********|*********|**
      character*23 LABEL(NSTAT)
      integer NSTAT, IDSTAT(NSTAT), IDSER(NSTAT)
      real LAT(NSTAT), LONG(NSTAT)

      open (unit=70,file='INPUT.dir/filelistno1', status = 'old')
      do 100 i = 1,NSTAT
        READ(70,*) IDSER(i),IDSTAT(i), LAT(i),LONG(i), LABEL(i)
 100  continue
      close(unit = 70, status = 'keep')
      RETURN
      END


      subroutine HTRCOUNT(X,N,NS,H,STR,STRNZ,TBMIN, ICOUNT,ZERO)
*********|*********|*********|*********|*********|*********|*********|**
*     written by Conan L. Hom
*     Thursday, October 31, 1996
*     This program counts the number of independent storms (ICOUNT) and
*     finds each storm's depth (H) and it's duration without the zero
*     hours removed (STR) and with the zero hours removed (STRNZ).  To
```

```
*    determine storm independence, Tbmin is given from a main program
*    and so is the actual matrix number N.
*********|*********|*********|*********|*********|*********|*********|**

      REAL X(N), H(NS), STR(NS), STRNZ(NS), TBMIN
*    Scan for first storm in timeline.  i.e. first precipitation
      do 100 I=1,N
        IF (X(I).gt.ZERO) THEN   !first precipitation encountered
          K = I
          GOTO 102
        endif
        IF (I.EQ.N.AND.X(I).LE.ZERO) THEN  !if no precip in period
          ICOUNT = 0
          GOTO 101              !goes to end of subroutine
        ENDIF
 100  CONTINUE
 101  CONTINUE
      GOTO 1000
 102  CONTINUE

*     storm counters
      HCOUNT = 0.0    !depth
      STRC = 0.0      !duration with periods of no precipitation
      STRNZC = 0.0    !duration without periods of no precipitation
      ICOUNT = 0      !counter of the number of storms
      GAP = 0.0       !counts duration of current gap of no precip.
      DO 200 I = K,N
*       adds to counters if storm gap was less than zero
        IF (X(I).GT.ZERO) THEN
          IF (GAP.LE.TBMIN) THEN
*           gap isn't greater than Tbmin thus add to current storm
              HCOUNT = HCOUNT + X(I)
              STRC= STRC + 1.  + FLOAT(GAP)
              STRNZC = STRNZC + 1.
              GAP = 0.
            ENDIF

*         records a storm if the gap is greater than tbmin, then resets
*           the counters
            IF (GAP.GT.TBMIN) THEN
              ICOUNT = ICOUNT + 1
              H(ICOUNT) = HCOUNT
              STR(ICOUNT) = STRC
              STRNZ(ICOUNT) = STRNZC

              HCOUNT = X(I)
              STRC = 1
              STRNZC =  1
              GAP = 0.
            ENDIF
          ENDIF

*         if zero hour then adds to gap
          IF (X(I).LE.ZERO) THEN
```

```
            GAP = GAP + 1.
         ENDIF
*        end of the matrix is reached
         IF(I.EQ.N) THEN
            ICOUNT = ICOUNT + 1
            H(ICOUNT) = HCOUNT
            STR(ICOUNT) = STRC
            STRNZ(ICOUNT) = STRNZC
            GOTO 201
         ENDIF
 200  CONTINUE
 201  CONTINUE

      GOTO 1011

 1000 DO 1010 I = 1,NS   !if no precipitation
         X(I) = 0.0000
         STR(I) = 0.0000
         STRNZ(I) = 0.0000
 1010 CONTINUE
 1011 CONTINUE
      RETURN
      END



      subroutine LEAP(year,months,nyear,lyear,NM,index,zero)
*********|*********|*********|*********|*********|*********|*********|**
*     leap determines the month layout for a (non)/leap year.
*     months  = number of days in a month.
*     lyear/nyear (leap and non leap no.  days in a month).
*     year = year being analyzed.
*          Conan L. Hom
*          October 28, 1996
*********|*********|*********|*********|*********|*********|*********|**

      integer year,months(NM), nyear(NM),lyear(NM)
      index = 1
      if (((float(year)/4.0)-(year/4)).lt.zero) then
         index = 2
      endif
      do 100 i1 = 1,NM
         if (index.eq.2) then
            months(i1) = lyear(i1)
         elseif (index.ne.2) then
            months(i1) = nyear(i1)
         endif
 100  continue
      RETURN
      END



      subroutine LINTERP1(xa, ya,x,y)
```

```
*********|*********|*********|*********|*********|*********|*********|**
*   Given two points (xa(1),ya(1)), (xa(2),ya(2)), subroutine finds
*   point (x,y) where x is between xa(1) and ya(2)) assuming that the
*   two data points are linearly related (y = mx+b) etc.
*   This assumes that x increases from xa(1) to xa(2).
*     linterp1 = linear interpolation in one dimension.
*     Conan L. Hom
*     Friday, July 18, 1997
*********|*********|*********|*********|*********|*********|*********|**
      real xa(2),ya(2), x, y

      dy = ya(2)-ya(1)
      dx = xa(2)-xa(1)
      y = ya(1) + (dy/dx)*(x-xa(1))
      RETURN
      END




      subroutine LINTERP2(x1a,x2a,ya,x1,x2,yint)
*********|*********|*********|*********|*********|*********|*********|**
*   Given arrays x1a(2) and x2a(2) and the function values ya(2,2) (i.e.
*   ya=ya(x1a,x2a)) this subroutine finds y for a point x1,x2 on the
*   x1, x2 plane.    It assumes that the ya is linearly related to x1 and
*   x2 and that x1 increases from x1a(1) to x1a(2) and x2 increases
*   from x2a(1) to x2a(2).    Basically a plane is formed in the y dir.
*   Yint is the final value
*     Conan L. Hom
*     Friday, July 18, 1997
*********|*********|*********|*********|*********|*********|*********|**
      real x1a(2), x2a(2),ya(2,2), x1, x2, yj(2), yint
      real y2a1(2)

      do 100 i = 1,2
         do 200 j = 1,2
            y2a1(j) = ya(i,j)
 200     continue
*          write(*,*) (ya(i,jj), jj = 1,2)
         call linterp1(x2a,y2a1,x2,yj(i))

*          write(*,*) yj(i)
 100  continue

      call linterp1(x1a,yj,x1,yint)
      RETURN
      END



      subroutine MAKEOUT(CDF,DEPTH,IR,IRNZ,ISE,ISTR,NAM,NS,TR,TRNZ)
*********|*********|*********|*********|*********|*********|*********|**
*     Makes the output files
*     Each output file is ISTR(i) x 6 (CDF,Depth,IR,IRNZ,TR,TRNZ)
```

```
*     Must figure out a way to record the ISTR
*         Conan L. Hom
*         Thursday, December 5, 1996
*********|*********|*********|*********|*********|*********|*********|**
      integer ISTR(ISE)
      real CDF(NS,ISE),DEPTH(NS,ISE),IR(NS,ISE),IRNZ(NS,ISE),TR(NS,ISE),
     $   TRNZ(NS,ISE)
      character*30 NAM(ISE)

      do 100 k = 1,ISE
         open(unit = 81,file = NAM(k),status = 'unknown')
         do 200 i = 1,ISTR(k)
            write(81,900) CDF(i,k),DEPTH(i,k),IR(i,k),IRNZ(i,k),TR(i,k),
     $         TRNZ(i,k)
 200     continue
         write(*,*) 'makout2'
         close(unit=81,status='keep')
 100  continue
 900  format(1x,f7.5,1x,f9.4, 2(1x,f8.4), 2(1x,f5.1))
      RETURN
      END



      subroutine NAMEIN(FNAME,IS,NF,NOFILE,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*       FNAME = Datafile to be read
*       Creates filename to be read
*       Uses a file called 'files' which contains all the series
*          numbers that are to be read.
*          Conan L. Hom
*          Friday, November 29, 1996
*********|*********|*********|*********|*********|*********|*********|**

      character*30 FNAME
      integer NSTAT,IS,NOFILE(NSTAT)

      open(unit = 91,file = 'INPUT.dir/files',status = 'old')
      do 100 i =1,NF
         read(91,*) NOFILE(i)
 100  continue
      close(unit = 91,status = 'keep')
      IFILE = NOFILE(IS)
      open(unit = 90,file = 'filenames',status = 'unknown')

      if (IFILE.lt.10) then
         write(90,1001) IFILE
      elseif (IFILE.ge.10.and.IFILE.lt.100) then
         write(90,1011) IFILE
      elseif (IFILE.ge.100.and.IFILE.lt.1000) then
         write(90,1021) IFILE
      endif

      close(unit = 90, status = 'keep')
```

```
      open(unit = 90, file = 'filenames', status = 'old')

      read(90,*) FNAME
      close(unit = 90, status = 'keep')

 1001 format('''SERIES.tst.dir/series',I1,'.tst''')
 1011 format('''SERIES.tst.dir/series',I2,'.tst''')
 1021 format('''SERIES.tst.dir/series',I3,'.tst''')
      RETURN
      END



      subroutine NAMEOUT(IS,ITB,ISE,NAM,NF,NOFILE,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*     Creates all the filenames to be written to.
*     NAM(1-4) = Spr,Sum,Fall,Wint output files
*     NOFILE = series numbers that will be read
*     IFILE = series number to be read currently
*     ITB = toggle 1 = CV, 2 = BP
*     This reads a file called 'files' which contains all the series
*        numbers to be read.
*        Conan L. Hom
*        Friday, November 29, 1996
*********|*********|*********|*********|*********|*********|*********|**

      character*30 NAM(ISE)
      character*2 TB
      integer IS,ISE,ITB,NOFILE(NSTAT)

      open(unit = 91,file = 'INPUT.dir/files',status = 'old')
      do 100 i = 1,NF
         read(91,*) NOFILE(i)
  100 continue
      close(unit = 91,status = 'keep')

      IFILE = NOFILE(IS)

      open(unit = 90, file = 'filenames',status = 'unknown')

      if (ITB.eq.1) TB = 'BP'
      if (ITB.eq.2) TB = 'CV'

      if (IFILE.LT.10) then
         write(90,1002) TB,IFILE
         write(90,1003) TB,IFILE
         write(90,1004) TB,IFILE
         write(90,1005) TB,IFILE
      elseif (IFILE.ge.10.and.IFILE.lt.100) then
         write(90,1012) TB,IFILE
         write(90,1013) TB,IFILE
         write(90,1014) TB,IFILE
         write(90,1015) TB,IFILE
      elseif (IFILE.ge.100.and.IFILE.lt.1000) then
```

```
            write(90,1022) TB,IFILE
            write(90,1023) TB,IFILE
            write(90,1024) TB,IFILE
            write(90,1025) TB,IFILE
         endif

         close(unit = 90, status = 'KEEP')
         open(unit = 90, file = 'filenames', status = 'old')

         do 200 k = 1,ISE
            read(90,*) NAM(k)
 200     continue

         close(unit = 90, status = 'KEEP')

 1002 format('''OUTPUT.dir/sprg',a2,I1,'.dat''')
 1003 format('''OUTPUT.dir/summ',a2,I1,'.dat''')
 1004 format('''OUTPUT.dir/fall',a2,I1,'.dat''')
 1005 format('''OUTPUT.dir/wint',a2,I1,'.dat''')

 1012 format('''OUTPUT.dir/sprg',a2,I2,'.dat''')
 1013 format('''OUTPUT.dir/summ',a2,I2,'.dat''')
 1014 format('''OUTPUT.dir/fall',a2,I2,'.dat''')
 1015 format('''OUTPUT.dir/wint',a2,I2,'.dat''')

 1022 format('''OUTPUT.dir/sprg',a2,I3,'.dat''')
 1023 format('''OUTPUT.dir/summ',a2,I3,'.dat''')
 1024 format('''OUTPUT.dir/fall',a2,I3,'.dat''')
 1025 format('''OUTPUT.dir/wint',a2,I3,'.dat''')
      RETURN
      END


      subroutine PWALLMAKE(PW,ISE,PWALL,NSTAT,IS,NOFILE,ISTAT)
*********|*********|*********|*********|*********|*********|*********|**
*     Conan L. Hom
*     Friday, July 26, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer ISE, NSTAT,IS, NOFILE(NSTAT), ISTAT
      real PW(ISE), PWALL(NSTAT,ISE+2)

      do 100 j = 1,ISE
         PWALL(IS,j + 2) = PW(j)
 100  continue
      PWALL(IS,1) = real(NOFILE(IS))
      PWALL(IS,2) = real(ISTAT)
      RETURN
      END


      subroutine PWENTER(LAT,LONG,PW,ISE,SEASONS)
*********|*********|*********|*********|*********|*********|*********|**
```

```
*  this subroutine will load in the correct Precipitable water data to
*  Be used in Normalization
*  Note the latitude must be within 89.5 N and 89.5 S.
*  The algorithm may need to be verified for S and E hemispheres.
*  This algorithm should be correct up to through thousands of a degree.
*  To swtich hemispheres you must uncomment and comment lines 261-264
*  Polint and linterp do the same thing it turns out.
*     Conan L. Hom
*     Monday, July 14, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer simp,ise
      parameter (m = 360, n = 180, ZER = 1.E-4, simp = 0)
      real LAT,LONG,PW(ISE)
      character seasons(ise)*4, PWfile*40
      real X1A(m),X2a(n),YA(2,2), X1, X2,X,Ct(2)
      integer itog(2), ii(2),ilow(2),ihi(2)
      real Xrhi(2),Xrlow(2),XA(2),YA1(2),GRID(m,n), XX1(m), XX2(n)


C-----Original Grid coordinates scaled back 0.5 degrees.
      do 3 i =1,m
         XX1(i) = real(i-1)

 3    continue

      do 4 i = 1,n
         XX2(i) = real(i-1)
 4    continue


C-----Station coordinates scaled back 0.5 degrees
C          Hemispehere dependent.
 261  Ct(1) = real(M-long)-0.5     !longitude (Western Hemisphere)
* 262 Ct(1) = real(long)-0.5       !longitude (Eastern Hemisphere)
 263  Ct(2) = real(n/2 - lat)-0.5  !latitude (N-Hemisphere)
* 264 Ct(2) = real(lat+n)-0.5      !latitude (S Hemisphere)
C          all cases include equator and 180 degree line....
      if(Ct(1).lt.0) Ct(1) = 360.+Ct(1)  !  360 degrees = 0 degrees
      if(abs(Ct(1)-360).lt.zer) Ct(1)=0.

      write(*,*) 'Scaled Station Coordinates',(Ct(i), i = 1,2)

      do 5 i = 1,2
         itog(i) = 0
         call ABOVEBELOW(Ct(i),Xrhi(i),Xrlow(i))
 5    continue

C-----simple method
      if (simp.eq.1) then
         do 6 i = 1,2
            itog(i) = 1
            dl = Ct(i)-Xrlow(i)
            dh = Xrhi(i)-Ct(i)
            if (dl.lt.dh) then
```

```
            ii(i) = int(Xrlow(i))
         elseif (dl.ge.dh) then
            ii(i) =int(Xrhi(i))
         endif
 6     continue
       goto 100
    endif

C-----Corners in fromt the grid to find the correct four data points
C-----surrounding the point....

    do 7 i = 1,2
       dl = Ct(i)-Xrlow(i)
       dh = Xrhi(i)-Ct(i)
       if (abs(dh).le.zer) then
          ii(i) = int(Xrhi(i))
          itog(i) = 1
       elseif (abs(dl).le.zer) then
          ii(i) = int(Xrlow(i))
          itog(i) = 1
       else
          itog(i) = 0
       endif
 7  continue
 100 continue

*     do 101 i = 1,2
*        write(*,*) Ct(i),Xrhi(i),Xrlow(i), ii(i), itog(i)
* 101 continue

    do 800 k = 1,4
C-------opening the file
       call PWFILECREATE(k,seasons,PWFILE,ISE)

C--------load PW data
       open(unit = 52, file = PWfile, status = 'old')
       do 810 i = 1,m
          read(52,*) (GRID(i,j), j= 1,n)
 810   continue
       close(unit = 52, status = 'keep')
       do 11 i = 1,2
          ilow(i) = int(Xrlow(i)+1.)
          ihi(i) = int(Xrhi(i)+1.)
 11    continue
       itest = itog(1) + itog(2)
       if (itest.eq.2) then
          PW(k) = GRID(ii(1)+1,ii(2)+1)   !if location falls on a point
       elseif (itest.eq.1) then      !longitude on line
          do 19 i = 1,2
             if (itog(i).eq.1) nq = i
 19       continue
          do 12 i = 1,2
             nn = 2
             qq = 0
```

```
            if (itog(i).ne.1) then
               write(*,*) ii(nq), ilow(i), ihi(i), nq
               X = Ct(i)   !X for polint
               if (i.eq.1) then
                  XA(1) = XX1(ilow(i))
                  XA(2) = XX1(ihi(i))
                  YA1(1) = GRID(ilow(i),ii(nq)+1)
                  YA1(2) = GRID(ihi(i),ii(nq)+1)
               elseif(i.eq.2) then
                  XA(1) = XX2(ilow(i))
                  XA(2) = XX2(ihi(i))
                  YA1(1) = GRID(ii(nq)+1,ilow(i))
                  YA1(2) = GRID(ii(nq)+1,ihi(i))
               endif
            endif
 12      continue
         write(*,*) 'prelinterp1',(YA1(qq),qq=1,2),XA(1),XA(2)

         call linterp1(XA,YA1,X,PW(k))

      elseif(itest.eq.0) then
         mm = 2
         nn = 2
         X1 = Ct(1)
         X2 = Ct(2)
         X1A(1) = XX1(ilow(1))
         X1A(2) = XX1(ihi(1))
         X2A(1) = XX2(ilow(2))
         X2A(2) = XX2(ihi(2))
         YA(1,1) = GRID(ilow(1),ilow(2))
         YA(2,2) = GRID(ihi(1),ihi(2))
         YA(1,2) = GRID(ilow(1),ihi(2))
         YA(2,1) = GRID(ihi(1),ilow(2))
*         write(*,*) x1, x2
*     write(*,*) 'linterp2-1',(YA(qq,1),YA(qq,2),X1A(qq),qq=1,2)
*     write(*,*) 'linterp2-2',(YA(1,qq),YA(2,qq),X2A(qq),qq=1,2)
         call linterp2(x1a,x2a,ya,x1,x2,PW(k))
      endif
 800  continue

 2001 format('''PW.dir/PWAvg_',A4,'.dat''')
      RETURN
      END



      SUBROUTINE PWFILECREATE(k,seasons,PWFILE,ISE)
*********|*********|*********|*********|*********|*********|*********|**
*     Conan L. Hom
*********|*********|*********|*********|*********|*********|*********|**
      integer k, ISE
      character seasons(ise)*4, PWfile*40

      open (unit = 51, file = 'scratch', status = 'unknown')
```

```
      write(51,1) seasons(k)
*      write(*,1) seasons(k)
      close(unit = 51, status = 'keep')
      open (unit = 51, file = 'scratch',status = 'old')
      read (51,*) PWfile
      close (unit = 51, status = 'keep')
 1    format('''PW.dir/PWAvg_',A4,'.dat''')
      RETURN
      END



      SUBROUTINE PWPRINT(ISE, PWALL,NF,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*     Conan L. Hom
*     Friday, July 25, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer ISE, NF, NSTAT
      real PWALL(NSTAT,ISE+2)

      open (unit = 52, file = 'statPW.dat', status = 'unknown')
      do i = 1,NF
         write(52,*) (PWALL(i,j), j= 1,ISE+2)
      enddo
      write(52,*) 'SERID, STID, SPRG, SUMM, FALL, WINT (mm)'

      close (unit = 52, status = 'keep')
      RETURN
      END



      subroutine SCRATCHFILE(IL,SFILE,N,ISTAT,NOFILE,IS,ICRUD)
*********|*********|*********|*********|*********|*********|*********|**
* Records file with defective data     1 = missing data -999
* 2 = missing data (as in the file ends early)
*      Conan L. Hom
*      Tuesday, July 29, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer N,ISTAT,NOFILE(N),IS,ICRUD, SFILE(N,3),ICRUD,IL

      if (IL.gt.0) then
         ICRUD = ICRUD+1
         SFILE(ICRUD,1) = NOFILE(IS)
         SFILE(ICRUD,2) = ISTAT
         SFILE(ICRUD,3) = IL
         close(unit = 10, status = 'keep')
      endif
      RETURN
      END



      subroutine SCRATCHPRINT(SFILE,NSTAT,ICRUD)
```

```
*********|*********|*********|*********|*********|*********|*********|**
*       Conan L. Hom
*       Tuesday, July 29, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer ICRUD,NSTAT,SFILE(NSTAT,3)
      character*30 REASON

      open(unit = 63, file = 'irtrduds.dat', status = 'unknown')
    do 100 i = 1,ICRUD
        if (SFILE(i,3).eq.2) then
          REASON = 'END OF FILE'
        elseif (SFILE(i,3).eq.1) then
          REASON = 'Missing DATA'
        endif
        write(63,1000) (SFILE(i,j),j=1,3),REASON
 100  continue
 1000 format (3(1x,I4),1x,A40)
      close(unit = 63, status = 'keep')
      RETURN
      END


      subroutine SEASONSTAT(C,D,IND,ISE,LYEAR,MONTHS,N,NH,NM,NY,NYEAR,
     $   P,S,SCOUNT,ST,S2,S3,S4,ZERO,ISTAT,NOFILE,NSTAT,NDATA,IS)
*********|*********|*********|*********|*********|*********|*********|**
*     S,ST,S2 represent Spr,Sum,Fall, Wint (i=1,4)
*     S = total rainfall in the season
*     ST = total hours in the season
*     S2 = total squared hourly rainfall in season
*     D(i,1) = mean rainfall for season i
*     D(i,2) = variance of rainfall for season i
*     D(i,3) = probability of no precipitation
*     D(i,4) = lag 1 autocorrelation
*     ints(i) = number of days in season-1
*       Conan L. Hom
*       Wednesday, November 20, 1996
*********|*********|*********|*********|*********|*********|*********|**

      real S(ISE),ST(ISE),S2(ISE), S3(ISE),S4(ISE),D(ISE,4),P(ISE)
      integer months(NM),nyear(NM),lyear(NM),INTS(4),index1,year1,
     $ SCOUNT(ISE),ISTAT,NSTAT,NSTAT,NOFILE(NSTAT),NDATA
      real C(N,NH)
*     discarding Jan and February 1971  (non leap)
      ifin = nyear(1)+nyear(2)

      do 50 k = 1,ISE
        S(k) = 0.
        ST(k) = 0.
        S2(k) = 0.
        SCOUNT(k) = 0
        S3(k) = 0
        S4(k) = 0
 50   continue
```

```
      do 99 jj = 1,2
        do 100 iy = 1,NY-1
          year1 = 1970+iy+1
          call leap(year1,months,nyear,lyear,NM,index1,zero)

*         season intervals
          ints(1) = nyear(3) + nyear(4) + nyear(5)
          ints(2) = nyear(6) + nyear(7) + nyear(8)
          ints(3) = nyear(9) + nyear(10) + nyear(11)
          if (index1.eq.1) ints(4) = nyear(12) + nyear(1) + nyear(2)
          if (index1.eq.2) ints(4) = nyear(12) + lyear(1) + lyear(2)

*         Statistics
          do 200 k = 1,ISE
            istar = ifin + 1
            if (k.eq.1) junk = istar
            ifin = ifin + INTS(k)
*            write(*,*) istar,ifin,ifin-istar+1
*            Drops seasons with missing data from stats
            if (NDATA.eq.2) then
 353            do 351 i = istar,ifin
                 do 352 j = 1,24
                   if (C(i,j).lt.0.or.C(i,j).gt.900) goto 200
 352              continue
 351            continue
            endif

            do 300 i =istar,ifin
              do 400 j = 1,24
                if (jj.eq.1) then
                  if (C(i,j).gt.zero) SCOUNT(k) = SCOUNT(k)+1
                  S(k) = S(k) + C(i,j)
                  ST(k) = ST(k) + 1.
                  S2(k) = S2(K) + C(i,j)*C(i,j)
                endif

C    First lag auto correlation numerator and denominator
                if (jj.eq.2) then
                  S4(k) = (C(i,j)-D(k,1))**2 + S4(k)
                  if (i.ne.ifin.and.j.ne.24) then
                    if (j.ne.24) then
                      crump = (C(i,j)-D(k,1))*(C(i,j+1)-D(k,1))
                      S3(k) = S3(k) + crump
                    endif
                    if(j.eq.24) then
                      crump = (C(i,j)-D(k,1))*(C(i+1,1)-D(k,1))
                      S3(k) = S3(k) + crump
                    endif
                  endif
                endif
 400          continue
 300        continue
 200      continue
```

```
*         write(*,*) 'total = ',ifin-junk+1
 100      continue
*       Calculate statistics (Mean, Variance, Pr(No Rain))

        do 500 k = 1,ISE
          if (jj.eq.1) then
            write(*,*) ST(k)
            D(k,1) = S(k)/ST(k)
            P(k) = D(k,1)
            D(k,2) = ((ST(k)*S2(k))-(S(k)*S(k)))/(ST(k)*(ST(k)-1))
            D(k,3) = 1.0 - float(SCOUNT(k))/ST(k)
          elseif (jj.eq.2) then
            D(k,4) = S3(k)/S4(k)
          endif
 500      continue
  99  continue

*       write(*,*) ind,ifin,(T(k)/24,k=1,ISE)
      write(44,*) Nofile(is)
      write(44,*) 'STID, Season, No.  Hrs, Mean, Variance,
     >         Prob(Mp Rain)'
      do 1000 k = 1,ISE
        write(44,*) ISTAT,k, ST(k), D(k,1), D(k,2), D(k,3)
 1000 continue
      RETURN
      END



      subroutine SEASONSTORM(C,DEPTH,H,IR,IRNZ,ISE,ISTART,ISTR,LYEAR,
     $  MINTB, N, NH, NM, NS, NY, NYEAR, STR, STRNZ, TR, TRNZ,X,ZERO,
     $  NDATA)
*********|*********|*********|*********|*********|*********|*********|**
*    SEASONSTORM separates the days and hours into seasons and then
*      counts the storms. Needs subroutine HTRCOUNT
*      istr is the total number of storms observed during the season
*
*      Conan L. Hom
*      Monday, November 25, 1996
*********|*********|*********|*********|*********|*********|*********|**

      real C(N,NH), DEPTH(NS,ISE), IR(NS,ISE), IRNZ(NS,ISE), TR(NS,ISE),
     $  TRNZ(NS,ISE), MINTB(ISE)
      integer ISTR(ISE), ISTART(ISE),INTS(4),NYEAR(NM),LYEAR(NM),
     $    MONTHS(12)
*     Needed for htrcount.f
      real X(N), TBMIN, H(NS), STR(NS), STRNZ(NS)

*     discarding Jan and February 1971 (non-leap)
      IFIN = NYEAR(1)+NYEAR(2)

*     Set total storm counter to zero
      do 50 k= 1,ISE
        ISTR(k) = 0
```

```
      ISTART(k) = 0
      INTS(k) = 0
 50   continue

    do 100 iy = 1,NY-1
      year1 = 1970+iy+1
      call LEAP(YEAR1,MONTHS,NYEAR,LYEAR,NM,INDEX1,ZERO)

*      season intervals
      INTS(1) = NYEAR(3) + NYEAR(4) + NYEAR(5)
      INTS(2) = NYEAR(6) + NYEAR(7) + NYEAR(8)
      INTS(3) = NYEAR(9) + NYEAR(10) + NYEAR(11)
      if (INDEX1.eq.1) INTS(4) = NYEAR(12) + NYEAR(1) + NYEAR(2)
      if (INDEX1.eq.2) INTS(4) = NYEAR(12) + LYEAR(1) + LYEAR(2)

      do 200 k = 1,ISE
        do 210 i = 1,N              !Zeroing out matrix
          X(i) = 0.
 210      continue

        ISTAR = IFIN+1              !Day range
        IFIN = IFIN + INTS(k)

*        Creation of data for htrcount
        NX = 24*INTS(k)             !Total number of hours
        do 300 i = ISTAR,IFIN       !Loading in rainfall
          do 400 j = 1,24
            X((i-ISTAR)*24+j) = C(i,j)
            if (NDATA.eq.2) then
              if (C(i,j).lt.0.or.C(i,j).gt.900) then
                ICOUNT = 0
                goto 359        !skips season with missing data
              endif
            endif
 400        continue
 300      continue

        TBMIN = MINTB(k)

        CALL HTRCOUNT(X,NX,NS,H,STR,STRNZ,TBMIN,ICOUNT,ZERO)
 359     continue
        IF (ICOUNT.EQ.0) GOTO 501

        ISTART(k) = ISTR(k)+1
        ISTR(k) = ISTR(k) + ICOUNT

*        records storms to season files.
*        creates storm intensity matrix too.
        do 500 i = ISTART(k),ISTR(k)
          DEPTH(i,k) = H(1+i-ISTART(k))
          TR(i,k) = STR(1+i-ISTART(k))
          TRNZ(i,k) = STRNZ(1+i-ISTART(k))
          IR(i,k) = DEPTH(i,k)/TR(i,k)
          IRNZ(i,k) = DEPTH(i,k)/TRNZ(i,k)
```

```
500        continue
501        continue
200     continue
100  continue
    RETURN
    END


    subroutine SERCHECK(C,fname,II,ind,lyear,months,N,nyear,NM,NH,NY,
   $  STID,ZERO,ISCRATCH,NDATA)
*********|*********|*********|*********|*********|*********|*********|**
* Checks for irregularities in the data (missing days, months, hours)
* C Contains the data for the station.
* STID (stat no, long, lat)
* II (year,month, day)
* NYEAR and LYEAR days per month during a normal and leap year
* Converts the stations readings from 1/100 inch to mm
*    Conan L. Hom
*    Monday, November 4, 1996
*********|*********|*********|*********|*********|*********|*********|**

    integer year,months(NM),NDATA
    integer nyear(NM),lyear(NM),II(N,3)
    real STID(3),C(N,NH)
    character*30 FNAME

    write(*,*) fname
*     data nyear /31,28,31,30,31,30,31,31,30,31,30,31/
*     data lyear /31,29,31,30,31,30,31,31,30,31,30,31/
    open(unit=10,file = fname,status = 'old')
    read(10,*) (STID(I),I = 1,3)
    ind = 0
    ISCRATCH = 0
    do 200 i1 = 1,NY
      year = 1970 + i1
      call leap (year,months,nyear,lyear,NM,index,zero)
      do 210 i2 = 1,NM
        do 220 i3 = 1,months(i2)
          ind = ind + 1
          read(10,*,END=241) (II(ind,q),q = 1,3),
   $                 (C(ind,j),j=1,NH)
          goto 243
241        if (i1.LT.NY) goto 242              !END OF FILE CHECK
          if (i1.eq.NY.and.i2.lt.3) goto 242   !END OF FILE CHECK
242        ISCRATCH = 2
          ICRUD = ICRUD+1
          CLOSE(unit = 10, status = 'keep')
          RETURN

243        do 240 j=1,NH                   !missing = zeros
244          if (C(ind,j).lt.0.or.C(ind,j).gt.900) then
              if (NDATA.eq.1) then      !fill in with zero
                C(ind,j) = 0.0
```

```
            endif
            ISCRATCH = 1
            ICRUD = ICRUD + 1
            if (NDATA.eq.0) then      !ditch the file
              close (unit = 10, status = 'keep')
              RETURN
            endif
          endif
          C(ind,j) = 0.254*C(ind,j)        !converts to INCHES here
240      continue

        if (II(ind,1).ne.year.or.
   $          II(ind,2).ne.i2.or.
   $          II(ind,3).ne.i3) then
          write(*,*) ind,year,i2,i3,(II(ind,qq),qq = 1,3),
   $            index
        endif
220    continue
210   continue
200  continue
     close(unit=10,status = 'keep')
     RETURN
     END



     subroutine STATIONFIND(IDSER,IDSTAT,IS,ISTAT,LABEL,LABSTAT,LAT,
   $    LATSTAT,LONG, LONGSTAT, NOFILE,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*    finds the corresponding station with the series id number
*    ISTAT = corresponding station number
*    Outputs the station information fo the station used.
*       Conan L. Hom
*       Monday, December 2, 1996
*********|*********|*********|*********|*********|*********|*********|**

     integer IDSTAT(NSTAT),IDSER(NSTAT),NOFILE(NSTAT),ISTAT
     real lat(NSTAT), latstat, long(NSTAT), longstat
     character*23 LABEL(NSTAT), LABSTAT

     ISTAT = 0
     Q = 0

     do 100 i = 1,NSTAT
       if (IDSER(i) .eq. NOFILE(IS)) then
         ISTAT = IDSTAT(i)
         LATSTAT = LAT(i)
         LONGSTAT = LONG(i)
         LABSTAT = LABEL(i)
       Q = 1
       endif
100  continue
     write(82,101) NOFILE(IS),ISTAT,LATSTAT,LONGSTAT,LABSTAT
101  format(1x, i3, 2x, I4, 2x, 2(f7.3,2x),'''',A23,'''',2x)
```

```
      if (Q.eq.0) write(*,*) 'ERROR----->could not find station!'
      RETURN
      END



      subroutine STATSMAKE(D,ISE,AVG,VARI,PRNO,LAG1,NSTAT,IS,
     >     NOFILE,ISTAT)
*********|*********|*********|*********|*********|*********|*********|**
*     Conan L. Hom
*     Friday, July 25, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer ISE, IS, ISTAT,NSTAT,NOFILE(NSTAT)
      real D(ISE,4), AVG(NSTAT,ISE+2), VARI(NSTAT,ISE+2),
     >     PRNO(NSTAT,ISE+2),LAG1(NSTAT,ISE+2)

      AVG(IS,1) = real(NOFILE(IS))
      VARI(IS,1) = real(NOFILE(IS))
      PRNO(IS,1) = real(NOFILE(IS))
      LAG1(IS,1) = real(NOFILE(IS))
      AVG(IS,2) = real(ISTAT)
      VARI(IS,2) = real(ISTAT)
      PRNO(IS,2) = real(ISTAT)
      LAG1(IS,2) = real(ISTAT)

      do i = 1,ISE
        AVG(IS,i+2) = D(i,1)
        VARI(IS,i+2) = D(i,2)
        PRNO(IS,i+2) = D(i,3)
        LAG1(IS,i+2) = D(i,4)
      enddo
      RETURN
      END



      subroutine STATSPRINT(ISE,AVG,VARI,PRNO,LAG1,NF,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*     Series #, ID #, spring, summer, fall, winter
*     PRNO = probability of no rain (PRNO)
*     VARN = variance (VARI)
*     LAG1 = lag 1 autocorrelation (LAG1)
*     AVRG = averages (AVG)
*     Conan L. Hom
*     Friday, July 25, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer ISE,NF, NSTAT
      real AVG(NSTAT,ISE+2), VARI(NSTAT,ISE+2), PRNO(NSTAT,ISE+2),
     $     LAG1(NSTAT,ISE+2)


      open(unit=54,file='OUTPUT.dir/statAVRG.dat', status = 'unknown')
      open(unit=55,file='OUTPUT.dir/statVARN.dat', status = 'unknown')
      open(unit=56,file='OUTPUT.dir/statPRNO.dat', status = 'unknown')
```

```
open(unit=57,file='OUTPUT.dir/statLAG1.dat', status = 'unknown')

do i = 1,NF
  write(54,*) (AVG(i,j),  j = 1,ISE+2)
  write(55,*) (VARI(i,j), j = 1,ISE+2)
  write(56,*) (PRNO(i,j), j = 1,ISE+2)
  write(57,*) (LAG1(i,j), j = 1,ISE+2)
enddo
do i = 1,4
  close(unit = 53+i, status = 'keep')
enddo

RETURN
END



subroutine STORMCDF(CDF,DEPTH,IR,IRNZ,ISE,ISTR,NS,TR,TRNZ,ZERO)
*********|*********|*********|*********|*********|*********|*********|**
*    Reorders the storms according to increasing storm depth (H)
*    If two or more storms have the same depth, it can assign one
*    TR, TRNZ combination to a depth....this is determined by the
*    first TR, TRNZ of the first of storms with the same depth.
*    CDF  Pr(X.lt.H)  (from lowest to highest H)
*    ISTR - total number of storms (NS max number of storms)
*      Conan L. Hom
*      Date unknown
*********|*********|*********|*********|*********|*********|*********|**

REAL DEPTH(NS,ISE),IR(NS,ISE), IRNZ(NS,ISE), TR(NS,ISE),
$    TRNZ(NS,ISE), CDF(NS,ISE), TEMP(10)
INTEGER ISTR(ISE)

*    Putting the Storms into order
do 100 k = 1,ISE
  do 200 i = 1,ISTR(k)-1
    do 300 j = 1,ISTR(k)-1
      if (DEPTH(j,k).gt.DEPTH(j+1,k)) then
          TEMP(1) = DEPTH(j,k)
        TEMP(2) = TR(j,k)
        TEMP(3) = TRNZ(j,k)
        TEMP(4) = IR(j,k)
        TEMP(5) = IRNZ(j,k)

        DEPTH(j,k) = DEPTH(j+1,k)
        TR(j,k) = TR(j+1,k)
        TRNZ(j,k) = TRNZ(j+1,k)
        IR(j,k)   = IR(j+1,k)
        IRNZ(j,k) = IRNZ(j+1,k)

        DEPTH(j+1,k) = TEMP(1)
        TR(j+1,k) = TEMP(2)
        TRNZ(j+1,k) = TEMP(3)
        IR(j+1,k) = TEMP(4)
```

```
            IRNZ(j+1,k) = TEMP(5)
          endif
300     continue
200   continue

*     Selecting only one duration for each storm depth
*       do 400 i = 2,ISTR(k)
*         if (abs(DEPTH(i,k)-DEPTH(i-1,k)).lt.ZERO) then
*           DEPTH(i,k) = DEPTH(i-1,k)
*           TR(i,k) = TR(i-1,k)
*           TRNZ(i,k) = TRNZ(i-1,k)
*           IR(i,k) = IR(i-1,k)
*           IRNZ(i,k) = IRNZ(i-1,k)
*         endif
* 400   continue

*     Computes reordered Storm data CDF
        do 500 i = 1,ISTR(k)
          CDF(i,k) = i/float(ISTR(k)+1)
500     continue
100   continue
      RETURN
      END


      subroutine STORMNORM(DEPTH,IR,IRNZ,ISE,ISTR,NS,P,PW,TR,TRNZ)
*********|*********|*********|*********|*********|*********|*********|**
*     This Normalizes IR,IRNZ by dividing by P.
*     Normalizes TR,TRNZ by dividing by (P/PW).
*     P = Seasonal average Preciptiation/Hr.
*     PW = Precipitable water.
*       Conan L. Hom
*       Monday, November 25, 1996
*********|*********|*********|*********|*********|*********|*********|**
      real DEPTH(NS,ISE), IR(NS,ISE),IRNZ(NS,ISE),P(ISE),PW(ISE),
     $     TR(NS,ISE),TRNZ(NS,ISE)
      integer ISTR(ISE)

      do 100 k = 1,ISE
        do 200 j = 1,ISTR(k)
          IR(j,k) = IR(j,k)/P(k)
          IRNZ(j,k) = IRNZ(j,k)/P(k)
          TR(j,k) = TR(j,k)/(P(k)/PW(k))
          TRNZ(j,k) = TRNZ(j,k)/(P(k)/PW(k))
200     continue
100   continue
      RETURN
      END


      subroutine TBMINLOAD(ISE,ISTAT,ITB,MINTB,NSTAT,TB,TBSTAT)
*********|*********|*********|*********|*********|*********|*********|**
```

```
*     Finds the MINTB matrix by searching each of the 'scat' files.
*     ITB determines whether it will give back a CV or a BP matrix
*        1 = BP, 2 = CV.
*        Conan L. Hom
*        Sunday, December 1, 1996
*********|*********|*********|*********|*********|*********|*********|**
      integer ISE,ISTAT,ITB,NSTAT,TBSTAT(NSTAT)
      real MINTB(ISE),TB(NSTAT,2)

      open (unit = 71,file = 'INPUT.dir/scatspr.dat',status = 'old')
      open (unit = 72,file = 'INPUT.dir/scatsum.dat',status = 'old')
      open (unit = 73,file = 'INPUT.dir/scatfall.dat',status = 'old')
      open (unit = 74,file = 'INPUT.dir/scatwin.dat',status = 'old')

      do 10 k= 1,ISE
        MINTB(k) = 0.0
 10   continue

      do 100 k = 1,ISE
        do 200 i = 1,NSTAT
          read(k+70,*) TBSTAT(i), (TB(i,j),j= 1,2)
 200    continue
        do 300 i = 1,NSTAT
          if (ISTAT.eq.TBSTAT(i)) THEN
            MINTB(k) = TB(i,ITB)
          endif
 300    continue
 100  continue

      close (unit = 71, status = 'keep')
      close (unit = 72, status = 'keep')
      close (unit = 73, status = 'keep')
      close (unit = 74, status = 'keep')
      RETURN
      END

C     END SUBROUTINES
```

Appendix F

PPF Selection

The following is the description and code of the program used to find the PPF points from the list of independent storms produced by program *IRTR.f*.

PPF.f and supporting Subroutines
Program Written by Conan L. Hom
Conceptualization Date: March 1997
Completion: July 1997
Revision for S.M. thesis Date: Wednesday, October 22, 1997
Language: Fortran77 PDQ
Supporting Subroutines in *ppfsubs.f*
File unit numbers used 50, 70, 80, 90, 91.

F.1 General Description

PPF.f takes the output produced from *IRTR.f* and selects storms for the production possibility frontier (PPF). The key subroutine for the selection process is GTBEFORE. The user has the flexibility to choose which data type to process. See lines 18–20 in the main program. Several inputs are noted:

1. iss is a 1 x 4 matrix of zeros or ones. A one indicates to process the data of the season. The positions of the matrix from left to right are spring, summer, fall,

and winter.

2. itb is a 1 x 2 matrix of zeros and ones. A one indicates to process the data from *IRTR.f* generated by the breakpoint method ((1,1) position) and/or the coefficient of variation method ((1,2) position).

3. nzeros is a 1 x 2 matrix. The (1,1) position indicates to process the storms with the hours of no rain included for computing the average intensity and duration. The (1,2) position indicates to prcocess the storms based without the hours of no rain included.

The output files are produced into a directory named PPFOUT.dir. The individual files are described in this example:

WZsummBP86.dat

- WZ = with zeros included (from nzeros). NZ = No zeros.
- summ = summer season (from iss).
- BP = breakpoint (from itb). CV = coefficient of variation.
- 86 = series number of the station.

The output the columns are

1. total storm depth [mm].
2. intensity i_r [mm/hr].
3. duration t_r [hr].
4. match (one if storm is actually a real data point, 0 otherwise). This is just a check to verify that the program worked. Match should have values of one.

For input, *PPF.f* needs the file 'PPF.in' in a directory marked 'INPUT.dir'. This file contains the series numbers of the stations that will be processed. The number of stations in this file needs to match the parameter NF in the main program (line

11). It also uses a file named filelistnol (in the INPUT.dir directory) which contains the series number, station ID number, latitude, longitude, and station name (in that order). The actual input data files are in a directory marked 'OUTPUT.dir.'

The general algorithm for GTBEFORE is described in the sample subroutine below conveniently named SAMPLE. We assume that INBIN is an N x 2 matrix with the first column as the intensity and the second column as the duration. In the general case, the columns could stand for whatever is on the X and Y axes. For this example, we assume further that the matrix has been ordered from highest intensity to lowest intensity - i.e. INBIN(1,1) is the highest intensity and INBIN(N,1) is the lowest. OUTBIN represents the points selected by the algorithm.

SAMPLE.f

```
      subroutine SAMPLE(N, INBIN,OUTBIN,ICKY)
      integer N,ICKY
      real INBIN(N,2), OUTBIN(N,2)
      zero = 1E-6  !threshold tolerance

*      select highest intensity storm as starting point for PPF
      do 100 j = 1,2
        OUTBIN(1,j) = INBIN(1,j)
 100  continue

      ICKY = 1 !count for no. points in OUTBIN)

      do 200 i = 2,N
        d1 = OUTBIN(icky,1) - INBIN(i,1) !compares intensities (ir)
        d2 = OUTBIN(icky,2) - INBIN(i,2) !compares durations  (tr)

           if (d1.gt.zero) then !next storm on list is of lower ir
             if (-d2.gt.zero) then !and tr of next storm is higher
               icky = icky+1   !storm combo selected
             do 300 j = 1,2
                 OUTBIN(icky,j) = INBIN(i,j)
C                  new point now to evaluate rest of storms on
 300           continue
          endif
          elseif(abs(d1).le.zero) then !case when ir's are equal
              if (-d2.gt.zero) then !if tr on next storm is higher
                 do 400 j = 1,2     !therefore replace the selected storm
                   OUTBIN(icky,j) = INBIN(i,j)
 400             continue
              endif
```

```
      elseif (d1.lt.(-1.*zero)) then
C           check if ir's actually ordered properly
          write(*,*) 'sort failed'
        endif
 100  continue
      RETURN
      END
```

The reader should note that there are two critical assumptions to the performance of this algorithm:

1. The slope or *rate of transformation* of the PPF, is always negative.

2. There is a one-to-one correspondance between the axes variables (intensity and duration).

These two assumptions ensure that the selected combinations characterize a PPF that is monotonicaly decreasing for higher duration and intensities. Moreover, the the second assumption creates the possibility of the data points to be characterized by a function.

F.2 Programs

F.2.1 Main Program: PPF.f

```
C234567
      program PPF1

*********|*********|*********|*********|*********|*********|*********|**
*     Conan L. Hom
*     Monday, July 28, 1997
*     Massachusetts Institute of Technology
*     Uses Splines
*********|*********|*********|*********|*********|*********|*********|**

      parameter (NF = 10, ISE = 4, NMAX = 1000, NSTAT =174, NBIN = 10)
      parameter (ZERO = 1E-4)

      integer iss(ISE), itb(2), nzeros(2), i0, i2a, i2

*     one indicates process for season or for tb type.
*     BP CV sprg summ fall wint
 18   data iss /0,1,0,0/
 19   data itb /1,0/
 20   data nzeros /1,0/


*********|*********| MATRICES CREATION
* STATIONLOAD
      character*23 LABEL(NSTAT)
      integer IDSER(NSTAT), IDSTAT(NSTAT)
      real LAT(NSTAT), LONG(NSTAT)

* SERIESIN
      integer INFILES(NSTAT)
* STATIONMATCH
      integer IDNO
* INOUT
      character*42 NAMIN, NAMOUT

* STORMCOUNTLOAD
*       integer IBP(NSTAT,7), ICV(NSTAT,7), IBPS(ISE), ICVS(ISE)
* DATALOAD
      real CDF(NMAX), H(NMAX),IR(NMAX), TR(NMAX)
      integer SCOUNT
* MAXTRIR
      real IRM, TRM
* BINTRIR
      real IRMBIN(NMAX,3), TRMBIN(NMAX,3)
      real IRINT(NBIN+1), TRINT(NBIN+1)
      integer XIRCOUNT(NMAX), XTRCOUNT(NMAX)
```

```
      real XIRBIN(NMAX,3), XTRBIN(NMAX,3)

*  ZELIM

* PPFPOINTS
      real POINTS(NMAX,3), SAVEPOINTS(NMAX,3)

* MODELS
      real PPFBIN(NMAX,4)
      integer icky

*    SORT/GTBEFORE
      real HS(NMAX),IRS(NMAX),TRS(NMAX)
      real XBIN(NMAX,3), GT1BIN(NMAX,3), GT2BIN(NMAX,3)
      integer IRorTR,icky1, icky2

*********|*********| END MATRICES CREATION
      do 100 i0 = 1,NF
        is = i0
        call STATIONLOAD (IDSER, IDSTAT,LABEL,LAT,LONG,NSTAT)
        call SERIESIN (INFILES, NF, NSTAT)
        call STATIONMATCH (IDSER,IDNO,INFILES,IS,NSTAT)
*         call STORMCOUNTLOAD (IBP,IBPS, ICV,ICVS,INFILES,IS,ISE,NSTAT)
*        processes BP or CV
        do 200 i1 = 1,2
          if (itb(i1).ne.1) goto 201
*          processes zeros or without zeros
          do 250 i2a = 1,2
            if (nzeros(i2a).ne.1) goto 251
*            processes chosen seasons
            do 300 i2 = 1,ISE
              if (iss(i2).ne.1) goto 301
              call INOUT(is,i1,i2,i2a,IDNO,IDSER,ISE,NAMIN,
     $             NAMOUT,NSTAT)
              write(*,*) 'ok'

              call DATALOAD(i1,i2a,i2,ISE,NAMIN,CDF,H,IR,TR,NMAX,
     $             SCOUNT)
              write(*,*) NAMIN
              call MAXTRIR(CDF,H,IR,IRM,NMAX,SCOUNT,TR,TRM)

              call BINTRIR(H, IR, IRM, IRMBIN, TR, TRM, TRMBIN,
     $             NBIN, NMAX, SCOUNT, XIRCOUNT, XTRCOUNT, IRINT,
     $             TRINT,  XIRBIN, XTRBIN,ZERO)

              write(*,*) NAMIN,NAMOUT,IRM, TRM


*    GREATER THAN BEFORE (ALONG BOTH IR and TR)
            IRorTR = 1
            call SORT(H,IR,TR,NMAX,SCOUNT,HS,IRS,TRS,IRorTR,XBIN)
            call GTBEFORE(NMAX,ICKY1,SCOUNT,GT1BIN,IRorTR,XBIN,
     $        ZERO)
            IRorTR = 2
```

```
        call SORT(H,IR,TR,NMAX,SCOUNT,HS,IRS,TRS,IRorTR,XBIN)
        call GTBEFORE(NMAX,ICKY2,SCOUNT,GT2BIN,IRorTR,XBIN,
     $   ZERO)
        call SAME(NMAX,ICKY1,ICKY2,GT2BIN,GT1BIN,POINTS,ZERO,
     $    ICKY)

      call ZELIM(POINTS,NMAX,ICKY,ZERO)
      call BINCHANGE(NMAX,POINTS,ICKY,SAVEPOINTS,ISAVE)

*            makes output
             call OUTMAKE(PPFBIN,NMAX,ICKY,POINTS,ZERO)
             call LABEL1(NMAX, PPFBIN, SAVEPOINTS, ICKY,
     $          ISAVE)

      write(*,*) 'OUTPUT ', icky
*            writes to file
             call OUTPUT(ICKY,PPFBIN,NMAX,NAMOUT)

 301         continue
 300       continue
 251       continue
 250     continue

 201       continue
 200     continue
 100   continue
      end
```

F.2.2 Supporting Subroutines: ppfsubs.f

```
*********|*********|*********|*********|*********|*********|*********|**
*********|*********|*********|*********|*********|*********|*********|**
*    SUPPORTING SUBROUTINES for PPF.f
*********|*********|*********|*********|*********|*********|*********|**
*********|*********|*********|*********|*********|*********|*********|**

      subroutine BINCHANGE(NMAX,INBIN,NIN,OUTBIN,NOUT)
*********|*********|*********|*********|*********|*********|*********|**
*     This is used to exchange names of bins (NMAX,3
*     Conan L. Hom (H, IR,TR)
*********|*********|*********|*********|*********|*********|*********|**
      integer NMAX,NIN,INBIN(NMAX,3),OUTBIN(NMAX,3),NOUT

      do 100 i = 1,NIN
        do 200 j = 1,3
          OUTBIN(i,j) = INBIN(i,j)
 200    continue
 100  continue
      NOUT = NIN
      RETURN
      END



      subroutine BINTRIR(H, IR, IRM, IRMBIN, TR, TRM, TRMBIN, NBIN,
     * NMAX, SCOUNT, XIRCOUNT, XTRCOUNT, IRINT, TRINT, XIRBIN, XTRBIN,
     * ZERO)
*********|*********|*********|*********|*********|*********|*********|**
*     This divides the IR and TR into NBIN number of segments and finds
*     the maximum TR and IR (respectively within each bin.
*     The IR-TR combos in each bin are in IRBIN (For all the TR's),
*       and TRBIN and the maximums are in the IRMBIN and TRMBIN, the
*       counts for each bin are int eh IR/TRCOUNTBIN
*       Note this subroutine is flexible in terms of changing bins
*       Conan L. Hom
*       Wednesday-Monday, May 28-June 2, 1997
*********|*********|*********|*********|*********|*********|*********|**
        integer NBIN, NMAX, SCOUNT
        integer XTRCOUNT(NMAX), XIRCOUNT(NMAX)
        real H(NMAX),IR(NMAX),TR(NMAX),IRINT(NBIN+1),TRINT(NBIN+1)
        real XIRBIN(NMAX,3),XTRBIN(NMAX,3),zero
        real IRM, IRMBIN(NMAX,3),TRM, TRMBIN(NMAX,3)


        XIR = IRM/10.
        XTR = TRM/10.

        IRINT(1) = ZERO
```

```
      TRINT(1) = ZERO
      IRINT(NBIN+1) = IRM
      TRINT(NBIN+1) = TRM

      do 100 i = 1,NBIN
         XTRCOUNT(i) = 0
         XIRCOUNT(i) = 0
         do 150 j = 1,3
            TRMBIN(i,j) = 0.0
            IRMBIN(i,j) = 0.0
 150     continue
 100  continue

      do 200 i = 1,NBIN-1
         IRINT(i+1) = IRINT(i) +XIR
         TRINT(i+1) = TRINT(i) +XTR
 200  continue

*     Put Storms into bins
      do 300 i = 1,NBIN
         do 400 j = 1,SCOUNT
            if(IR(j).gt.IRINT(i)) then
               if (IR(j).le.IRINT(i+1)) then
                  XIRCOUNT(i) = XIRCOUNT(i) + 1
                  q = XIRCOUNT(i)
                  XIRBIN(q,1) = H(j)
                  XIRBIN(q,2) = IR(j)
                  XIRBIN(q,3) = TR(j)
               endif
            endif
            if(TR(j).gt.TRINT(i)) then
               if(TR(j).le.TRINT(i+1)) then
                 XTRCOUNT(i) = XTRCOUNT(i)+1
                  q = XTRCOUNT(i)
                  XTRBIN(q,1) = H(j)
                  XTRBIN(q,2) = IR(j)
                  XTRBIN(q,3) = TR(j)
               endif
            endif
 400     continue

*        finds maximums
         do 500 j = 1,XTRCOUNT(i)
            if (XTRBIN(j,2).gt.TRMBIN(i,2)) then
               do 550 k = 1,3
                  TRMBIN(i,k) = XTRBIN(j,k)
 550           continue
            endif
 500     continue
         do 600 j = 1,XIRCOUNT(i)
            if (XIRBIN(j,3).gt.IRMBIN(i,3)) then
               do 650 k = 1,3
                  IRMBIN(i,k) = XIRBIN(j,k)
 650           continue
```

```
          endif
 600     continue
 300  continue
      RETURN
      END


      subroutine DATALOAD(i1,i2a,i2,ISE,NAMIN,CDF,H,IR,TR,NMAX,SCOUNT)
*********|*********|*********|*********|*********|*********|*********|**
*     This loads the PDF, storm depth and IR, TR (TBCV and NZ,WZ)
*        specified
*      Conan L. Hom
*     Tuesday, May 13, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer i1,i2a,i2,ISE,NMAX,SCOUNT
      character*42 NAMIN
      real CDF(NMAX),H(NMAX),IR(NMAX),TR(NMAX), unk1, unk2

      open(unit = 91, file = NAMIN,status = 'old')

*     with zeros
      if (i2a.eq.1) then
         do 100 i = 1,NMAX
            read(91,*,END = 101) CDF(i), H(i), IR(i), unk1, TR(i), unk2
 100     continue
 101     continue
      endif

*     without zeros
      if(i2a.eq.2) then
         do 200 i = 1,NMAX
            read(91,*,END = 201) CDF(i), H(i), unk1, IR(i), unk2, TR(i)
 200     continue
 201     continue
      endif
      SCOUNT = i-1
      close(unit = 91, status = 'keep')

*      do 300 i = 1,SCOUNT
*         write(*,*) H(i), IR(i), TR(i), ' loading'
* 300  continue
*      pause
      RETURN
      END


      subroutine GTBEFORE(NMAX,ICKY,SCOUNT,OUTBIN,IRorTR,INBIN,ZERO)
*********|*********|*********|*********|*********|*********|*********|**
*     This goes along one axis and only selects if the value on the
*     axis is greater than the one before it.
*     This assumes that the subroutine SORT has been run on the data
*     Thus the data is ordered from high to low for a given IRorTR
```

```
*     (1 = IR, 2 = TR).
*     Conan L. Hom
*     Tuesday, June 3, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer ICKY, IRorTR, SCOUNT, NMAX, iq
      real OUTBIN(NMAX,3),INBIN(NMAX,3),ZERO

      if (IRorTR.eq.1) iq = 3
      if (IRorTR.eq.2) iq = 2
      k = IRorTR + 1

*     select highest intensity storm as starting point for PPF
      do 200 j = 1,3
         OUTBIN(1,j) = INBIN(1,j)
 200  continue
      icky = 1          !count for number of points in OUTBIN

      do 300 i = 2,SCOUNT
         d1 = OUTBIN(icky,k)-INBIN(i,k)
         d2 = OUTBIN(icky,iq)-INBIN(i,iq)
         if (d1.gt.ZERO) then
            if (-d2.gt.ZERO) then
               icky = icky+1   !storm combo selected
               do 400 m = 1,3
                  OUTBIN(icky,m) = INBIN(i,m)
C                new point now to evaluate rest of storms on
 400           continue
            endif
         elseif (abs(d1).le.zero) then
            if (-d2.gt.ZERO) then
               do 450 m = 1,3
                  OUTBIN(icky,m) = INBIN(i,m)
 450           continue
            endif
         elseif (d1.lt.(-1.*ZERO)) then
C             check if sorted properly
            write(*,*) 'sort failed'
         endif
 300  continue
      RETURN
      END


      subroutine INOUT(is,i1,i2,i2a,IDNO,IDSER,ISE,NAMIN,
     $    NAMOUT,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*      creates OUTPUT FILE PPF series no.  method cv or bp season
*      creates data files to be specified for processing.
*      NAMIN, NAMOUT
*     Conan L. Hom
*     Thursday, May 8, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer is, i1, i2, IDNO,ISE, NSTAT, IDSER(NSTAT),iser
```

```
      character*42 NAMIN, NAMOUT
      character*2 TB,ZE
      character*4 SEASON

      iser = IDSER(IDNO)

      open (unit = 90, file ='filenames', status = 'unknown')

      if (i1.eq.1) TB = 'BP'
      if (i1.eq.2) TB = 'CV'
      if (i2a.eq.1) ZE = 'WZ'
      if (i2a.eq.2) ZE = 'NZ'
      if (i2.eq.1) SEASON = 'sprg'
      if (i2.eq.2) SEASON = 'summ'
      if (i2.eq.3) SEASON = 'fall'
      if (i2.eq.4) SEASON = 'wint'

      if (ISER.lt.10) then
         write(90,1001) SEASON,TB,ISER
         write(90,1002) ZE,SEASON,TB,ISER
      elseif (ISER.ge.10.and.ISER.lt.100) then
         write(90,1003) SEASON,TB,ISER
         write(90,1004) ZE,SEASON,TB,ISER
      elseif (ISER.ge.100.and.ISER.lt.1000) then
         write(90,1005) SEASON,TB,ISER
         write(90,1006) ZE,SEASON,TB,ISER
      elseif (ISER.ge.1000) then
         write(*,*) 'Error, need to design Subroutine INOUT to handle'
         write(*,*) 'station series no.  greater or equal to 1000'
         stop
      endif

      close (unit = 90, status = 'keep')
      open (unit = 90, file = 'filenames', status = 'old')
      read(90,*) NAMIN
      read(90,*) NAMOUT
      close (unit = 90, status = 'keep')

 1001 format('''OUTPUT.dir/',A4,A2,I1,'.dat''')
 1002 format('''PPFOUT.dir/',A2,A4,A2,I1,'.dat''')
 1003 format('''OUTPUT.dir/',A4,A2,I2,'.dat''')
 1004 format('''PPFOUT.dir/',A2,A4,A2,I2,'.dat''')
 1005 format('''OUTPUT.dir/',A4,A2,I3,'.dat''')
 1006 format('''PPFOUT.dir/',A2,A4,A2,I3,'.dat''')
      RETURN
      END


      subroutine LABEL1(NMAX,OUTBIN,INBIN,IOUT,INN)
*********|*********|*********|*********|*********|*********|*********|**
*     This just labels points on the PPF which are actual data
*     points (can be used for labeling) or weighting actual data points
*     OUTBIN(i,4) = 1 if match or 0 if not
```

```
*********|*********|*********|*********|*********|*********|*********|**
*     This just labels points on the PPF which are actual data
*     Conan L. Hom
*     Monday, July 7, 1997
*     ZERO1 is used for comparing rainfall measurements.....
*********|*********|*********|*********|*********|*********|*********|**
      parameter (ZERO1 = 0.0059)
      integer NMAX, IOUT,INN
      real INBIN(NMAX,3), OUTBIN(NMAX,4)

      do 100 i = 1,IOUT
         OUTBIN(i,4) = 0.
         do 200 j = 1,INN
            icount = 0
            do 300 k = 1,3
               del = abs(OUTBIN(i,k)-INBIN(j,k))
               if(del.le.zero1) icount = icount+1
 300        continue
            if (icount.eq.3) then
               OUTBIN(i,4) = 1.
               write(*,*) 'yay', i,j
               goto 201
            endif
 200     continue
 201     continue
 100  continue
      RETURN
      END



      subroutine MAXTRIR(CDF,H,IR,IRM,NMAX,SCOUNT,TR,TRM)
*********|*********|*********|*********|*********|*********|*********|**
*     Finds the maximum value IR, and TR values (IRM,TRM) from the
*     storm list
*     Conan L. Hom
*     Wednesday, May 28, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer SCOUNT, NMAX
      real CDF(NMAX), H(NMAX), IR(NMAX), TR(NMAX), IRM, TRM

      IRM = 0.0
      TRM = 0.0
      do 100 i = 1,SCOUNT
         if (IR(i).gt.IRM) IRM = IR(i)
         if (TR(i).gt.TRM) TRM = TR(i)
 100  continue
      RETURN
      END



      subroutine OUTMAKE(PPFBIN,NMAX,N,INBIN,ZERO)
*********|*********|*********|*********|*********|*********|*********|**
```

```
*     Makes the output IR,TR, H output PPF for the various other methods
*     Conan L. Hom
*     Thursday, June 12, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer N, NMAX
      real PPFBIN(NMAX,4),ZERO,INBIN(NMAX,3)

      do 100 i = 1,N
         do 150 j = 1,3
            PPFBIN(i,j) = INBIN(i,j)
 150     continue
 100  continue
      RETURN
      END



      subroutine OUTPUT(N, INBIN, NMAX, NAMOUT)
*********|*********|*********|*********|*********|*********|*********|**
*     OUTPUTS PPF (H, IR, TR) onto file NAMOUT
*     Conan L. Hom
*     Sunday, June 8, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer NMAX, N
      real INBIN(NMAX,4)
      character*42 NAMOUT

      open(unit = 50, file = NAMOUT, status = 'unknown')
      do 100 i = 1,N
         if (INBIN(i,1).gt.ZERO) then
            write(50,*) (INBIN(i,j), j= 1,4)
         endif
 100  continue
      close(unit = 50, status = 'keep')
      RETURN
      END



      subroutine SAME(NMAX,ICKY1,ICKY2,PPF2BIN,PPF1BIN,PPFBIN,ZERO,ICKY)
*********|*********|*********|*********|*********|*********|*********|**
* Finds common points between the two methods of GT than before.
* As usual, ICKY = total stormcount of the PPFBIN.
*     Conan L. Hom
*     Wednesday, June 11, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer NMAX, ICKY, ICKY1,ICKY2
      real PPF2BIN(NMAX,3),PPF1BIN(NMAX,3), PPFBIN(NMAX,3), ZERO
      real diff(3)

      icky = 0
      do 100 i = 1, ICKY1
         do 200 j = 1,ICKY2
            icount = 0
```

```
          do 300 k= 1,3
             diff(k) = PPF1BIN(i,k)-PPF2BIN(j,k)
             if (abs(diff(k)).lt.zero) then
                icount = icount + 1
             endif
 300      continue
          if(icount.eq.3) then
             icky = icky+1
             do 400 k= 1,3
                PPFBIN(icky,k)= PPF1BIN(i,k)
 400         continue
             goto 100
          endif
 200    continue
 100  continue
      call ZELIM(PPFBIN,NMAX,ICKY,ZERO)
      RETURN
      END



      subroutine SERIESIN(INFILES,NF,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*     Loads input series numbers
*     INFILES(input series numbers)
*     Conan L. Hom
*     Thursday, May 8, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer NSTAT, INFILES(NSTAT),NF

      open(unit = 91, file = 'INPUT.dir/PPF.in', status = 'old')
      do 100 i = 1,NF
         read(91,*) INFILES(i)
 100  continue
      close (unit = 91, status = 'keep')
      RETURN
      END



      subroutine SORT(H,IR,TR,NMAX,SCOUNT,HS,IRS,TRS,IRorTR,XBIN)
*********|*********|*********|*********|*********|*********|*********|**
*     Sorts H, IR, TR, according to high to low IR or TR
*     IRorTR:  1 = sort by IR, 2 = sort by TR
*     OUTPUT: HS, IRS, TRS
*     Conan L. Hom
*     Tuesday, June 3, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer NMAX,SCOUNT,IRorTR
      real H(NMAX),HS(NMAX),IR(NMAX),IRS(NMAX),TR(NMAX),TRS(NMAX)
      real XBIN(NMAX,3), SCRAP(3)

      q = IRorTR + 1
```

```
      do 100 i = 1,SCOUNT
         XBIN(i,1) = H(i)
         XBIN(i,2) = IR(i)
         XBIN(i,3) = TR(i)
 100  continue

      do 200 i = 1,SCOUNT-1
         do 300 j = 1, SCOUNT-1
            if (XBIN(j+1,q).gt.XBIN(j,q)) then
               do 350 k = 1,3
                  SCRAP(k) = XBIN(j,k)
                  XBIN(j,k) = XBIN(j+1,k)
                  XBIN(j+1,k) = SCRAP(k)
 350           continue
            endif
 300     continue
 200  continue

      do 400 i = 1,SCOUNT
         HS(i) = XBIN(i,1)
         IRS(i) = XBIN(i,2)
         TRS(i) = XBIN(i,3)
 400  continue
      RETURN
      END


      subroutine STATIONLOAD(IDSER,IDSTAT,LABEL,LAT,LONG,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*     Loads in the series numbers with the corresponding stations
*     IDSTAT (list of all the station numbers
*     IDSER (list of all the series numbers)
*      IDSTAT(i) corresponds with IDSER(i) for all i.
*     Conan L. Hom
*     Thursday, May 8, 1997
*********|*********|*********|*********|*********|*********|*********|**

      character*23 LABEL(NSTAT)
      integer NSTAT, IDSTAT(NSTAT), IDSER(NSTAT)
      real LAT(NSTAT),LONG(NSTAT)

      open (unit = 70, file = 'INPUT.dir/filelistno1', status = 'old')
      do 100 i = 1,NSTAT
         read(70,*) IDSER(i), IDSTAT(i), LAT(i), LONG(i), LABEL(i)
 100  continue
      close(unit = 70, status = 'keep')
      RETURN
      END


      subroutine STATIONMATCH(IDSER,IDNO,INFILES,IS,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
```

```
*     Extracts the station INFO of station being processed.
*     does this by simply finding which IDSER is being used...
*     i.e.  IDSER(IDNO)
*     Conan L. Hom
*     Friday, May 9, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer NSTAT,IDNO,IS, IDSER(NSTAT), INFILES(NSTAT)

      IDNO = 0
      do 100 i = 1,NSTAT
         if (INFILES(is).eq.IDSER(i)) IDNO = i
 100  continue
      if (IDNO.eq.0) then
         write(*,*) ('Error, station not on list')
         stop
      endif
      RETURN
      END


      subroutine STORMCOUNTLOAD(IBP,IBPS, ICV,ICVS,INFILES,IS,ISE,NSTAT)
*********|*********|*********|*********|*********|*********|*********|**
*     Loads in storm count data for the available stations.
*     This is an optional subroutine but it can be used to compare
*     if the counts are correct....
*     Conan L. Hom
*     Monday, May 12, 1997
*********|*********|*********|*********|*********|*********|*********|**
      integer NSTAT, ISE, IS, INFILES(NSTAT)
      integer IBP(NSTAT,7), ICV(NSTAT,7), IBPS(ISE), ICVS(ISE)

      open (unit = 80, file = 'OUTPUT.dir/stormcount', status = 'old')

      icount = 0
      do 100 i = 1,NSTAT
         read(80,*,END = 101) (IBP(i,j),j= 1,7)
         read(80,*,END = 101) (ICV(i,j),j= 1,7)
 100  continue
 101  continue
      close (unit = 80, status = 'keep')
      icount = i-1

      do 150 ix = 1,4
         ICVS(ix) =  0
         IBPS(ix) = 0
 150  continue

      do 200 i2 = 1,icount
         if (INFILES(IS).eq.IBP(i2,1)) then
            do 300 i3 = 1,4
               ICVS(i3) = ICV(i2,3+i3)
               IBPS(i3) = IBP(i2,3+i3)
 300        continue
```

```
          goto 201
        endif
 200  continue
 201  continue
      RETURN
      END


      subroutine ZELIM(ZERBIN,NMAX,ICKY,ZERO)
*********|*********|*********|*********|*********|*********|*********|**
* Eliminates storms of zero depth from the plots of the PPF
*     Conan L. hom
*     Wednesday, June 11,1997
*********|*********|*********|*********|*********|*********|*********|**
      integer icky, NMAX,itot
      real ZERBIN(NMAX,3),ZERO

      itot = 0
      do 100 i = 1,ICKY
         if (abs(ZERBIN(i,1)).gt.zero) then
         itot = itot+1
            do 200 k = 1,3
               ZERBIN(itot,k) = ZERBIN(i,k)
 200        continue
         endif
 100  continue
      icky = itot
      RETURN
      END
```

Bibliography

[1] Baldassare Bacchi, Gianfranco Becciu, and Nath. T. Kottegoda. Bivariate exponential model applied to intensities and durations of extreme rainfall. *Journal of Hydrology*, 155:225–236, 1994.

[2] Rafael L. Bras. *Hydrology: An Introduction to Hydrologic Science.* Addison-Wesley Publishing Co., Reading, 1990.

[3] R. H. Brooks and A. T. Corey. Properties of porous media affecting fluid flow. *Journal of Irrigation and Drainage Division, American Society of Civil Engineering*, 92((IR2)):61–88, 1966.

[4] Carlo Cao. A contribution to statistical depth-duration-frequency analysis. *Journal of Hydrology*, 22:109–129, June 1974.

[5] National Climate Data Center. Hourly Precipitation Records. Updated Anually, 1989.

[6] Morris H. DeGroot. *Probability and Statistics.* Addison-Wesley Publishing Co., Reading, second edition, 1989.

[7] Peter S. Eagleson. Dynamics of flood frequency. *Water Resources Research*, 8(4):878–898, August 1972.

[8] Peter S. Eagleson. Cimate, soil, and vegetation 1. introduction to water balance dynamics. *Water Resources Research*, 14(5):705–712, October 1978.

[9] Peter S. Eagleson. Climate, soil and vegetation 2. the distribution of annual precipitation derived from observed storm sequences. *Water Resources Research*, 14(5):713–721, October 1978.

[10] Dara Entkehabi, November 1996. Lecture from 1.71 Introduction to Hydrology.

[11] Dara Entkehabi, April 1997. Lectures from 1.711 Engineering Hydrology.

[12] Dara Entkehabi, March 1997. Lectures from 1.711 Engineering Hydrology.

[13] David L. Randel et al. Nasa Vapor Project Data Set (NVAP). National Aeronautics and Space Administration, July 1995.

[14] Sun K. Hom, December 1997. In person discussion *en route* to Lexington, MA.

[15] George J. Huffman, Robert F. Adler, Phillip Arkin, Alfred Chang, Ralph Ferraro, Arnold Gruber, John Janowiak, Alan McNab, Bruno Rudolf, and Udo Schneider. The Global Precipitation Climatetology Project (GPCP) Combined Precipitation Dataset. *Bulletin of the American Meteorological Society*, 78(1):5–19, January 1997.

[16] George J. Huffman, Robert F. Adler, Bruno Rudolf, Udo Schneider, and Peter R. Keehn. Global precipitation estimates based on a technique for combining satellite-based estimates, rain gauge analysis, and nwp model precipitation information. *Journal of Climate*, 8:1284–1295, May 1995.

[17] Justin E. Kerwin. Introduction to Geometric Modelling and Computation. 13.016 Lecture Notes, August 1994.

[18] R.S. Kurothe, N. K. Goel, and B. S. Mathur. Derived flood frequency distribution for negatively correlated rainfall intensity and duration. *Water Resources Research*, 33(9):2103–2107, September 1997.

[19] Marco Marani, Giovanna Grossi, Francesco Napolitano, Michael Wallace, and Dara Entekhabi. Forcing, intermittency, and land surface hydrologic partitioning. *Water Resources Research*, 33(1):167–175, January 1997.

[20] Steven A. Margulis. Temporal disaggregation of monthly precipitation estimates obtained from earth-orbiting satellites for use in hydrologic applications. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, January 1998.

[21] Walter Nicholson. *Microeconomic Theory: Basic Principles and Extensions.* Dryden Press, New York, fifth edition, 1992.

[22] Robert S. Pindyck and Daniel L. Rubinfeld. *Economic Models and Economic Forecasts.* McGraw-Hill, Inc., New York, third edition, 1991.

[23] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in Fortran: The Art of Scientific Computing.* Cambridge University Press, New York, second edition, 1992.

[24] Pedro J. Restrepo-Posada and Peter S. Eagleson. Identification of independent rainstorms. *Journal of Hydrology*, 55:303–319, 1982.

[25] Ignacio Rodriguez-Iturbe. Scale of fluctuation of rainfall models. *Water Resources Research*, 22(9):15S–37S, AUGUST 1986.

[26] Ignacio Rodriguez-Iturbe, Vijay K. Gupta, and Ed Waymire. Scale considerations in the modeling of temporal rainfall. *Water Resources Research*, 20(11):1611–1619, November 1984.

[27] G. D. Salvucci and Dara Entekhabi. Equivalent steady soil moisture profile and the time compression approximation in water balance modeling. *Water Resources Research*, 30(10):2737–2749, October 1994.

[28] Vijay P. Singh. *Elementary Hydrology.* Prentice Hall, Inc., Englewood Cliffs, 1992.

[29] Juan B. Valdes, Ignacio Rodriguez-Iturbe, and Vijay K. Gupta. Approximations of temporal rainfall from a multidimensional model. *Water Resources Research*, 21(8):1259–1270, August 1985.

[30] Joan Walker, October 1997. Electronic Mail.

[31] Wayne. L. Winston. *Operations Research.* Wadsworth Publishing Company, Belmont, 1994.

[32] Jennifer K. Wynn. Intensity-Frequency-Duration curves for three cities in the U. S. Unpublished Paper: Correspond with Darae Entekhabi, July 1993.

[33] Jennifer K. Wynn. Seasonal and geographic variability in rainstorm parameter distributions. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1994.